UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Clasificador de Emociones mediante Análisis de Imágenes

Víctor Cendrós Capdevila
Tutor: José María Martínez Sánchez

JULIO 2020

Clasificador de Emociones mediante Análisis de Imágenes

AUTOR: Víctor Cendrós Capdevila
TUTOR: José María Martínez Sánchez

Video Processing and Understanding Lab
Dpto. Tecnología Electrónica y de las Comunicaciones Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2020

Resumen (castellano)

Este Trabajo Fin de Grado se basa en el estudio del reconocimiento facial orientado a la detección y clasificación de emociones, con el fin de desarrollar un programa que sea capaz de realizar dicha tarea mediante la extracción de características faciales de interés en un flujo de vídeo grabado con cualquier dispositivo de captura de vídeo.

Se podrá ver el trabajo pasado referido a reconocimiento facial que ha desembocado en las herramientas con las que hoy podemos contar para investigar en temas derivados del reconocimiento facial, como en este caso el reconocimiento de emociones.

El trabajo desempeñado se ha ejecutado en dos vertientes, siendo la primera la investigación del tema a tratar, el entendimiento de los músculos faciales, los gestos que son capaces de generar y su implicación directa con las emociones; y la segunda vertiente siendo la creación de una aplicación capaz de clasificar un total de 6 emociones a partir de la detección de la cara de un sujeto grabado en vídeo del cual se extraen características faciales.

Palabras clave (castellano)

Reconocimiento facial, detección, clasificación, emoción, sentimiento

Abstract (English)

This Bachelor Thesis is based on the current study of facial recognition oriented to emotion detection and classification, with the purpose of developing a software able to do the mentioned task by extracting facial characteristics of interest via video flux previously recorded with any recording device.

Past work related to facial recognition will be addressed, which has led to many tools and frameworks we can count on today to investigate topics derived from facial recognition, such as emotion recognition in this specific case.

The work done in this thesis has been carried out in two aspects, the first one being the investigation of the subject to be treated, the understanding of facial muscles, the gestures that they are capable of generating and their direct involvement with emotions; and the second aspect being the creation of an application capable of classifying a total of 6 emotions from the detection of the face of a subject recorded on video from which facial characteristics are extracted.

Keywords (inglés)

Facial recognition, detection, classification, emotion, sentiment

Agradecimientos

A nivel académico, quisiera agradecer a José María su gran labor como tutor de este trabajo, ha sabido guiarme de manera espléndida a lo largo de este trabajo y se ha involucrado en el proceso, manteniendo reuniones de manera constante, lo cual aprecio enormemente debido a su interés por que yo aprendiese a lo largo de este proyecto.

En el ámbito más personal, quisiera agradecer a todos aquellos que me han rodeado y me han acompañado en este largo y provechoso trayecto.
De corazón quiero daros las gracias por animarme cuando más lo he necesitado, por estar simplemente a mi lado en momentos tanto buenos como menos buenos.

Pero sobre todo me gustaría resaltar dos nombres propios: Alberto y Nuria. Mis padres.

Gracias por haberme permitido hacer lo que en un principio pensaba que quería, aquello en lo que a mitad de trayecto me planteé si verdaderamente era lo que deseaba estudiar, y ahora en el final puedo decir sin ninguna duda que es definitivamente aquello que quería hacer. Gracias por volcaros en mi educación, tanto académica como no. Me habéis ofrecido ayuda hasta en cosas que al principio no podíais, pero hacíais y hacéis el esfuerzo por aprender y poder ayudarme, y eso es algo por lo que estoy inmensamente agradecido.
Gracias por estar siempre que os necesitaba, fuera para lo que fuera.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

Con el paso de los años y el desarrollo de las máquinas de captación de imágenes y vídeos, el interés por crear aplicaciones o funcionalidades relacionadas con el análisis de vídeo ha crecido a la par notablemente. Concretamente, el interés por analizar comportamientos faciales de las personas ha incrementado sustancialmente.

Biológicamente, los seres humanos estamos diseñados de tal manera que, en cualquier interacción, social la cara es la parte del cuerpo que mayor influencia tiene sobre el resto de seres humanos, debido a que a través de ella únicamente, somos capaces de distinguirnos del resto gracias a las expresiones faciales.

Concretamente, aquello que nos diferencia del resto de seres vivos, a parte del raciocinio, es la capacidad de poder expresar sentimientos mediante gestos faciales.

El ser humano posee la capacidad de realizar y distinguir numerosas emociones desde el nacimiento a pesar de cambios en el estímulo visual tales como el efecto de envejecimiento en la vista u otros cambios en otros seres humanos como la presencia de gafas o vello facial en la cara de otro sujeto.

Es por esta causa principalmente por la que he realizado este estudio y desarrollo, por la peculiaridad del tema y por conocer más a fondo la tecnología detrás de las técnicas para reconocimiento y detección de emociones.

1.2 Objetivos

El objetivo principal de este Trabajo de Fin de Grado es el de crear una aplicación capaz de reconocer emociones mediante la detección de gestos faciales gracias a la captura de un fichero de vídeo, cuyas características han de ser extraídas por un programa externo de extracción de características, para finalmente poder clasificar en base a los resultados obtenidos la emoción reconocida.

Además de esto, los objetivos secundarios son el estudio del estado del arte, desde los inicios hasta la actualidad conociendo los cambios, los descubrimientos más relevantes que propiciaron el interés tan grande que actualmente tiene este campo de investigación, la búsqueda de *datasets* válidos para la finalidad que se busca y por último ,y no por ello menos relevante, una toma de contacto con el mundo de la investigación y el desarrollo de aplicaciones o funcionalidades que tienen como base el reconocimiento facial y sus múltiples variantes, que sin temor a errar se puede afirmar que va a ser uno de los campos donde más atención se va a enfocar la investigación tecnológica en un futuro no muy lejano.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Introducción
- Estado del arte

- Diseño de la aplicación
- Integración, pruebas y resultados
- Conclusiones y Trabajo Futuro

2 Estado del arte

2.1 Introducción

En este capítulo se tratarán los orígenes del reconocimiento facial, base en la que se apoya la detección de emociones que es el grueso de este trabajo, así como la tecnología en constante desarrollo y mejora que permite que se puedan llevar a cabo estos experimentos. Además, se explicará a nivel superficial el caso de una aplicación real de detección de emociones, mostrando resultados y técnicas que se emplean para desempeñar tal función.

2.2 Reconocimiento Facial

2.2.1 Orígenes

El estudio de detección de características faciales tiene su origen en la inquietud por descubrir técnicas de reconocimiento facial, una de las líneas más consolidadas del reconocimiento biométrico, fiables las cuales puedan garantizar un porcentaje de éxito muy elevado, idealmente de casi el 100% con un margen de error de décimas. Las primeras aproximaciones que se tomaron en este tema fueron las de crear estos sistemas con algoritmos como SIFT (*Scale–Invariant Feature Transform*)[1], LBP (*Local Binary Patterns*) [2] o Vectores de Fisher [3].

Una de estas aproximaciones, como ejemplo, fue el uso de *Eigenfaces* [4] para el reconocimiento facial. Esta técnica se basa principalmente en el uso de *sets*, conjuntos, de *Eigenvectors*, que son vectores (realmente matrices en el caso de imágenes) que cambian debido a factores de escala cuando una transformación lineal se aplica sobre ellos. Explicándolo de manera burda, a estas *Eigenfaces* se las puede considerar como caras estándar las cuales combinadas linealmente dan lugar a una cara que queremos identificar. Estos conjuntos de vectores representan los principales componentes que forman una cara. La figura siguiente muestra un pequeño conjunto de *Eigenfaces*.

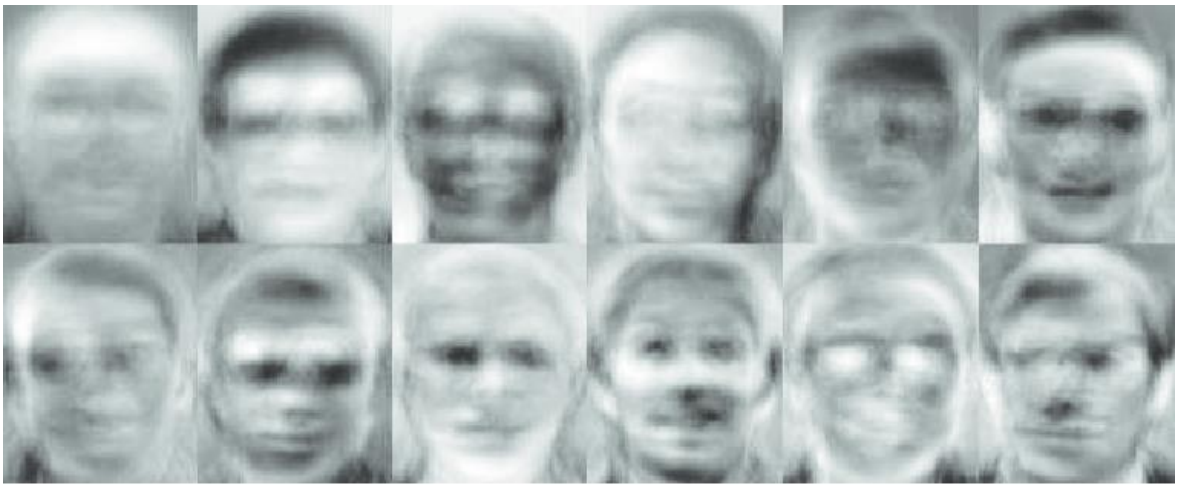

Figura 2-1 Ejemplo de conjunto de *Eigenfaces* (imagen extraída de [5])

Teniendo este set de *Eigenfaces*, se puede entonces determinar los *Eigenvectors* y *Eigenvalues* de la matriz de covarianza que forma este conjunto de imágenes.

2.2.2 Últimas tendencias

Hasta el año 2011 la tendencia y la preferencia para el reconocimiento facial era la de usar estos algoritmos anteriormente mencionados de manera conjunta y en todas sus variantes, pero Le et al. [6] a finales ese mismo año demostraron que mediante la utilización de un *autoencoder*, entrenado mediante Descenso por Gradiente Estocástico (SGD) en 1000 máquinas usando datos de vídeos de la plataforma YouTube, se conseguían resultados mejores que con los métodos utilizados hasta ese momento. Luego en 2012, A. Krizhevskt et al. [7] avanzaron en la investigación y se concluyó que redes neuronales similares a la propuesta en 2011 se podían entrenar con un ordenador y dos GPUs, reduciendo de esta manera la infraestructura necesaria para poder entrenar una Red Neuronal Densa (DNN).

Este hito marcó el inicio del *Deep Learning* aplicado a la extracción, y también clasificación, de características en imágenes como el algoritmo de-facto. De esta manera, se puede afirmar que gracias a las técnicas de *Deep Learning* y a la existencia de bases de datos inmensas se ha conseguido mejorar tanto la algoritmia que supera incluso a la capacidad humana en esta tarea.

Actualmente para la extracción de características faciales no se utiliza única y exclusivamente estas Redes Neuronales Densas, sino que tienen que estar a su vez acompañados previamente de otros algoritmos como la "Frontalización 3D" empleada en la herramienta *DeepFace* de Facebook, el cual recrea un modelo 3D de la cara en la imagen bajo estudio para luego, mediante el uso de unas transformaciones afines, "frontalizar" la imagen, es decir, hacer como si la imagen original hubiera sido tomada con la persona de la imagen teniendo su cara frente a la cámara, mirando directamente hacia ella.

2.2.3 Reconocimiento Facial con modelos 3D.

A lo largo de las últimas cuatro décadas se ha investigado y avanzado mucho en las técnicas de reconocimiento facial en dos dimensiones (2D) usando imágenes como datos de entrada a los sistemas, pero sin embargo el reconocimiento facial sigue siendo un gran reto en circunstancias de cambios de pose, de iluminación parcial o global debido a que una imagen es una proyección de una cara en tres dimensiones en la que una de estas dimensiones, la profundidad, se ha perdido.

A pesar de no ser tan frecuentes como las cámaras que captan imágenes en dos dimensiones, la tecnología detrás de los dispositivos de captura digital 3D ha avanzado notablemente, siendo más barata cada vez, pero sin rebatarle el protagonismo a la captación 2D.
Los datos faciales en 3D representan puntos en tres dimensiones dando lugar a formas geométricas tridimensionales, que son datos que con captación 2D no se tenían, por tanto, pueden ayudar a solventar los problemas mencionados anteriormente.

Brevemente se detallarán los métodos que se usan para el reconocimiento facial con esta tecnología, que son [19]:

- *Facial curve based*: este método se basa en extraer curvas faciales representativas de los datos faciales 3D como características. El problema de este método está en la heterogeneidad de los esquemas para desarrollar y llevar a cabo tal extracción

de curvas. Al final del proceso de este método, se definen unas métricas para comparar las uniones de las curvas con unas curvas faciales de referencia,

- *Shape descriptor based*: este método se basa en definir descriptores de forma, diseñados previamente para ser invariantes a rotaciones con el propósito de ser codificados de manera robusta. Las partes rígidas de la cara se extraen las primeras ya que van a ser las que menos cambio experimenten en cuanto a forma, mientras que las partes de la cara más flexibles, con mayor probabilidad de cambio con el tiempo, como puede ser la boca, se extraen y se codifican de otra manera distinta. Estas formas entonces se localizan en el llamado "*local shape map*" [10] para poder entonces definir la zona facial al completo.
- *Holistic matching based*: con este método se consigue reconocer la cara mediante "face surfaces" a las que se les aplica el algoritmo ICP (Iterative Closest Point) y la distancia de Hausdorff.
- *Prominent regions based*: en este método, las regiones prominentes de la cara son detectadas para formar vectores de características. Se usa el algoritmo AdaBoost para seleccionar aquellas regiones que fueran más representativas. Según propusieron Zhong et al. [11], se calculan "*Gabor features*" de cada región 3D seleccionada para conseguir crear un *Learned Visual Codebook* (LVC) basado en los resultados de un *clustering K–means* de los vectores de características. El reconocimiento facial es conseguido con un clasificador de vecino más cercano.

2.3 Reconocimiento de emociones

Si bien es cierto que la mayoría de la investigación hasta el momento se centra particularmente en el desarrollo de técnicas las cuales sean capaces de mejorar la parte algorítmica del problema que aún existe en reconocimiento facial, hay estudios que llevan estas herramientas de detección y extracción de características faciales a otros ámbitos de aplicación, como bien puede ser la detección de emociones.

Este tema en concreto es uno en el que no se está trabajando a gran escala, pues todavía existen dudas de la viabilidad de aplicaciones destinadas a la detección de emociones mediante expresiones faciales debido a que al ser humano aún le es difícil distinguir y reconocer con certeza absoluta las distintas emociones que puede ser capaz otro ser humano de reflejar. Este hecho se debe principalmente a que cada individuo es único, por tanto, su manera de reflejar emociones lo es también.

Todos los sistemas cuya tarea es reconocer o detectar emociones tienen una estructura muy parecida, lo que suele denominarse una estructura canónica. Esto se debe a que es una estructura que desde el inicio de la investigación en este tema se hizo de tal manera, haciendo que el orden de operación no se cambiase a pesar de los avances en el campo de investigación porque es la única manera coherente y con funcionalidad asegurada. Esta estructura se explica en el apartado siguiente siguiendo el caso de una de las arquitecturas más populares.

2.3.1 Caso concreto: FILTWAM

Existen algunos estudios que persiguen generar un software con el que poder detectar emociones mediante flujos de vídeo, como puede ser el caso del *framework* FILTWAM [8] el cual es capaz de reconocer en tiempo real emociones mediante el uso de una

webcam. Este *framework* no se sustenta única y exclusivamente en el flujo de vídeo capturado por la *webcam*, sino que también es capaz de detectar emociones mediante estímulos vocales recogidos por un micrófono. En este *paper* sin embargo se limitan a ofrecer y detallar los resultados obtenidos mediante el reconocimiento de emociones por estímulo visual.

Para llevar a cabo esta tarea, FILTWAM consta de un componente capaz de extraer características faciales para posteriormente poder reconocer y categorizar las emociones. Este componente, a su vez, se puede dividir en otros tres sub–componentes, cada uno de ellos encargado de realizar una tarea determinada. Estos sub–componentes que trabajan de manera consecutiva, y en el orden que a continuación se lista, son los siguientes:

- *Face detection*: reconocimiento de la cara del sujeto. El proceso tiene que tener su comienzo en este paso, ya que hay que localizar la cara antes de proceder a clasificar. Se especifica que se busca localizar la cara, pero no reconocer la cara en particular de un sujeto, es decir, se busca encontrar una cara para estudiarla, no se busca la cara de alguien en concreto ni se busca una cara para asociarla a un nombre.
- *Facial feature extraction*: extracción de características faciales relevantes con el fin de poder trabajar en base a ellos en el último sub–componente
- *Facial emotion classification*: clasificador de emociones. Este componente lleva a cabo la labor de analizar secuencias de vídeo, extrayendo una imagen por cada fotograma del vídeo que se quiere examinar.

La Tabla 2-1 muestra la matriz de confusión que los propios investigadores adjuntan en su artículo.

		Recognized Emotion by the Software							
		Happy	Sad	Surprise	Fear	Disgust	Angry	Neutral	Total
Requested Emotions	Happy	88	2	6	6	10	1	7	120
		73.4%	1.7%	5%	5%	8.3%	0.8%	5.8%	100%
	Sad	0	46	5	8	10	9	12	90
		0%	51.1%	5.6%	8.9%	11.1%	10%	13.3%	100%
	Surprise	0	0	60	6	4	0	10	80
		0%	0%	75%	7.5%	5%	0%	12.5%	100%
	Fear	0	7	6	40	11	7	9	80
		0%	8.8%	7.5%	50%	13.8%	8.7%	11.2%	100%
	Disgust	3	5	1	6	63	8	4	90
		3.3%	5.6%	1.1%	6.7%	70%	8.9%	4.4%	100%
	Angry	0	2	2	3	12	59	2	80
		0%	2.5%	2.5%	3.7%	15%	73.8%	2.5%	100%
	Neutral	4	15	52	19	10	5	355	460
		0.9%	3.2%	11.3%	4.1%	2.2%	1.1%	77.2%	100%
Total		95	77	132	88	120	89	399	1000

Tabla 2-1 Matriz de confusión del framework FILTWAM (extraída de [8])

A raíz de inspeccionar esta tabla, podemos inferir que el tema de reconocimiento y clasificación de emociones es un ámbito más complejo que el reconocimiento facial. Según estos investigadores, el porcentaje de éxito que asegura este software es del 72%, muy lejos del ideal 100% con el que se trabaja de manera aproximada en reconocimiento facial.

Sin embargo, este 72% es una cifra razonable debido a la complejidad del problema, sabiendo sobre todo que una persona es capaz de percibir y distinguir una emoción de manera correcta en el 67.67% de los casos, fijándonos en los datos que se proporcionan en este artículo. Si profundizamos en estos dos porcentajes, parece bastante difícil de asimilar que un programa sea capaz de determinar con más precisión que una persona la emoción que está mostrando un ser humano, ya que los humanos somos aquellos que reflejamos estas emociones y quienes enseñamos a un programa cuáles son las diferencias de estas emociones entre sí.

Esta confusión se puede aclarar explicando el método para conseguir estos porcentajes. En este estudio se contó con 10 voluntarios para que cada uno de ellos hiciera 100 expresiones faciales distintas, correspondientes con las emociones de la Tabla 2-1. Los investigadores pidieron a cada uno de los 10 voluntarios que hicieran 12 expresiones de felicidad, 9 de tristeza, 8 de sorpresa, 8 de miedo, 8 de repugnancia, 8 de enfado y otras 46 de expresión neutra. Estas imágenes se etiquetaron de esta manera, sin importar inicialmente la capacidad de cada voluntario de reflejar de manera exacta la emoción que se pedía. Cada una de estas personas, a las que se les sumaron otros dos "*raters*" del propio estudio, tenía luego que juzgar las imágenes que representaban cada una de las emociones, un total de un millar de fotos. El software entrenado y los voluntarios junto con los "*raters*" dieron los resultados anteriormente mencionados. Cabe destacar que los voluntarios no eran necesariamente actores, por tanto, su capacidad de imitar emociones bajo demanda no era excepcional, así como su capacidad de diferenciar emociones no estaba tampoco entrenada.

La problemática de esto reside en la capacidad de los individuos de mostrar emociones mediante la combinación Micro- y Macro-expresiones faciales [20]. Principalmente, la diferencia entre ambas es la intensidad y la duración en el tiempo de los movimientos de los músculos faciales: las Micro-expresiones son de una intensidad muy baja y tiempo muy pequeño, mientras que las Macro-expresiones son aquellas con intensidad de movimiento muscular notable y de una duración más prolongada.

2.3.2 Métodos para la detección de expresiones faciales

Conociendo entonces la diferencia entre estos dos tipos de expresiones faciales, podemos entonces comprender los resultados que arrojan los métodos utilizados hasta el momento para la extracción de expresiones faciales.

La Tabla 2-2 muestra las diferencias entre el porcentaje de éxito de extracción de expresiones dependiendo de si nos encontramos en el caso de estudio de Macro – expresiones o Micro – expresiones.
Esta tabla lista una serie de técnicas distintas, que son:

- LBP [2]: *Local Binary Patterns*. Representación eficiente de imágenes basado en texturas. La imagen bajo estudio se divide entre numerosas secciones, se extraen de ahí las características y se concatenan en un histograma de características para ser posteriormente utilizado como descriptor facial.

- PHOG [12]: *Pyramid of Histograms of Orientation Gradients*. El objetivo es representar una imagen por su forma y por su diseño espacial (*spatial layout*). La forma se consigue representar gracias a unos histogramas de la orientación de los bordes en una sub - región de la imagen cuantizada en *K bins.* Se consigue hacer una pirámide de distintas resoluciones HOG, quedando finalmente el descriptor PHOG como una concatenación de todos los vectores HOG de cada uno de los niveles de la pirámide.

TABLE 1
State-of-the-art methods for macro and micro expressions (* data augmentation).

Based on	Macro expression (CK+)		Micro expression (CASME II)	
App.	LBP [29] Block-based	90.05%	LBP [21] Block-based	55.87%
	PHOG [7] Salient region	95.30%	HIGO [21] Block-based	67.21% magnified
	CNN [8] Whole face	96.76% *	CNN [20] Whole face	47.30% *
Geom.	Gabor Jet [30] Facial points	95.17%	/	/
	DTGN [16] Facial points	92.35% *	/	/
Motion	LBP-TOP [12] Block-based	**96.26%**	DiSTLBP-IIP [23] Block-based	64.78%
	Optical flow [31] Facial meshes	93.17%	MDMO [19] Facial meshes	**67.37%**
	CNN + LSTM [14] Whole face	**98.62% ***	CNN + LSTM [24] Whole face	60.98% *

Tabla 2-2 Resultados de detección de emociones con Macro – expresiones o Micro – expresiones (extraída de [9])

- HIGO [21]: *Histograms of Image Gradient Orientation*. Se trata de una variante del PHOG. A diferencia del mencionado anteriormente, este reduce la influencia de la iluminación y el contraste en una imagen ignorando la magnitud de la primera derivada, por lo tanto, en situaciones reales en entornos no controlados, HIGO tiene un rendimiento mejor al PHOG.
- *Gabor Jet* [13]: Los *Gabor jets* son colecciones de coeficientes de Gabor de la misma localización en una imagen. Estos coeficientes son generados usando wavelets de distintas orientaciones, frecuencias y tamaños. La función principal de los *Gabor jets* es la de refinar los *landmarks* faciales, que se explicarán más adelante.
- DTGN [14]: *Deep Temporal Geometry Network.* Se basa en una red neuronal densa completamente conectada. Su función es la de extraer características geométricas temporales extraídas de *landmarks*.
- Flujo óptico (*Optical Flow*) [15]: Estos métodos de estimación sirven para caracterizar dinámicas locales faciales de texturas temporales, reduciendo el estudio de estas últimas a un análisis de una secuencia de patrones móviles. Este método es cuestionado debido a que la precisión decae enormemente si existen discontinuidades o cambios de iluminación.

- DiSTLBP-IIP [16]: *Discriminative Spatiotemporal Local Binary Pattern with an Improved Integral Projection*. Técnica usada únicamente en Micro – expresiones en la que se preserva la forma de los atributos en Micro – expresiones, se incorpora una proyección integral con operadores LBP en dominios espacial y temporal. Para discriminar entre Micro – expresiones se utiliza un método basado en la laplaciana.
- MDMO [17]: *Main Directional Mean Optical Flow*. Usado para el reconocimiento de Micro – expresiones espontáneas. Se detecta la cara y se divide en regiones de interés, operando sobre el flujo de vídeo en cada fotograma. Basándose en esa región de interés y en el alineamiento facial se propone un vector de características por región de 72 elementos, que sirven para entrenar un clasificador SVM.
- CNN + LSTM [18]: Se trata de una técnica en la que se combinan Redes Neuronales Convolucionales con *Long–Short Term Memory*. (LSTM) [22] es una red recurrente diseñada para solucionar los problemas de *Back–Flow* gracias a un algoritmo basado en el gradiente.

2.3.3 Estudio de Macro y Micro expresiones: conceptos y resultados

Al igual que les acontece a los seres humanos, toda herramienta desarrollada para la detección de Macro–expresiones tiene un porcentaje de acierto grande, superando en todos los casos expuestos en la tabla el 90%, mientras que en el estudio de las Micro–expresiones la probabilidad de éxito en la predicción baja sustancialmente, no superando en ningún caso ni siquiera el 70% de acierto.
Ambas sin embargo son comparables debido a que el clasificador final es el mismo, empleándose así unas máquinas de vectores de soporte (SVMs) con una validación cruzada de 10 grupos, la más común.

Estudiando la tabla, observamos que hay tres categorías diferentes que representan los modelos en los que están basados: aparición, geometría y movimiento. Nos centraremos única y exclusivamente en los modelos basados en geometría debido a que ha sido precisamente el tipo de modelo empleado en el desarrollo de este trabajo. Sin embargo, para mencionar algo de los otros dos modelos, podemos sobre todo destacar la naturaleza estática de los modelos basados en apariencia, es decir, que se estudia cada *frame* como caso particular sin tener en cuenta una aproximación temporal tal y como se hace en algunos modelos de geometría y en todos los de movimiento. Además de tener en cuenta el flujo temporal, los modelos basados en movimiento tienen también fundamentos de texturas dinámicas, tema en el que no entraremos en esta memoria.

Los modelos basados en geometría se sustentan en la localización de puntos de interés en las caras presentes en los vídeos bajo estudio. Estos puntos no son aleatorios, se busca situarlos en zonas de la cara relevantes para un futuro estudio, pudiendo ser estas localizaciones los extremos de los ojos, los bordes exteriores de los labios, el contorno de las cejas, etc.

A partir de esto se puede inferir precisamente el porqué de que las herramientas de detección de expresiones basados en modelos geométricos son buenos a nivel de Macro–expresiones, ya que debido a los movimientos más intensos de la musculatura facial al ejecutar este tipo de expresiones se experimentan cambios apreciables en la localización de estos puntos de interés, conocidos técnicamente como "*landmarks*". Por el contrario, en el caso de las Micro–expresiones podemos por el mismo motivo decir que no es deseable

utilizar este método debido a que las posiciones de estos puntos de interés no se van a ver lo suficientemente desplazadas, dándose incluso el caso de que no exista desplazamiento alguno, como para poder decidir si el sujeto ha realizado o no un cambio de expresión facial (se puede ver en la Tabla 2-2 que no se mencionan métodos geométricos en el estado del arte aplicados a Micro-expresiones).

2.3.4 *Action Units*: concepto y clasificación

Para finalizar con el tema de los *landmarks*, podemos introducir con la información ofrecida el concepto de *Action Units* (AU), que son precisamente las características usadas en mayor medida en este Trabajo de Fin de Grado.

Las Action Units (AU) son un estándar para, simplemente, describir expresiones faciales. El sistema *Facial Action Coding System* (FACS), desarrollado originalmente por el anatomista Carl–Herrman Hjotsjö y publicado por Paul Ekman, Wallace V. Friesen y Joseph C. Hager en 1978, es aquel encargado de clasificar movimientos faciales por su apariencia en la cara. Evidentemente, para que esto sea posible hace falta tener en cuenta la naturaleza temporal y continua de los flujos de vídeo ya que un movimiento se caracteriza por la diferencia de posición de puntos en la cara entre dos instantes de tiempo, de ahí que el *Facial Action Coding System* sea capaz de codificar movimientos de los músculos faciales y saber asociarlos a las *Action Units* que procede.

A pesar de ser ésta la definición estricta de *Action Units*, se verá más adelante cuando se explique el funcionamiento de la aplicación diseñada, no es necesario tener en cuenta la continuidad en el tiempo del vídeo, es decir, es posible extraer las *Action Units* a una imagen estática o a un fotograma en específico. Esto se debe a que no solo se es capaz de extraer estas AUs por el movimiento, sino también por la configuración de los puntos de interés de una cara.

En total, hay una lista de *Action Units* formada por 42 diferentes, que se diferencian entre ellas dependiendo del gesto realizado y de los músculos empleados para tal acción. A ellas se les nombra mediante la notación "*AU_XX*", siendo la variable "*XX*" el número correspondiente a la *Action Unit*. En la página web de la compañía "*iMotions*" [https:2//imotions.com/], que es una compañía fundada en 2005 con el ánimo de fomentar el estudio y comercialización de aplicaciones capaces de detectar emociones, así como muchas otras aplicaciones biométricas, se pueden visualizar unos vídeos que muestran las acciones realizadas en cada *Action Unit*.

Estas AUs están distribuidos de la siguiente manera:

- **AU_01, AU_02, AU_04**: Relacionado con el movimiento de cejas.
- **AU_05, AU_06, AU_06, AU_07**: Relacionado con el movimiento de los párpados.
- **AU_09 a AU_14**: Relacionado con el movimiento de la zona de la nariz y de las mejillas.
- **AU_15 a AU_18, AU_20 y AU_22 a AU_28**: Relacionado con el movimiento de los labios superior e inferior y de la barbilla.
- **AU_41 a AU_46**: Relacionado con la oclusión o no de los párpados.
- **AU_51 a AU_58**: Relacionado con la posición de la cabeza, por ejemplo, si está inclinada a derecha, izquierda o hacia atrás.
- **AU_61 a AU_64**: Relacionado con el movimiento de los globos oculares.

En la siguiente imagen podremos ver algunos de estas AUs.

Figura 2-2 Ejemplo visual de las AUs 04, 09, 42 y 52 (Imágenes extraídas de https://imotions.com/blog/facial-action-coding-system/)

Como último detalle de las *Action Units*, pueden ser descritas no únicamente mediante su presencia o ausencia de ella, lo cual es una descripción objetiva, sino también con la intensidad, puntuada de 0 a 5 en valores continuos. Esta última manera de describirlas, a diferencia de la presencia, no es algo objetivo del todo sino parcialmente subjetivo ya que la intensidad no se puede evaluar de manera tan exacta, ni tampoco existen tablas ni parámetros concretos con los que contrastar estas intensidades y sentenciar que una sea mínima o máxima.

3 Diseño

Esta sección tiene como propósito mostrar cuál es el proceso que se lleva a cabo dentro de la aplicación para conseguir detectar y clasificar emociones, así como explicar cada módulo que la compone.

3.1 Diseño funcional de la aplicación

Esta aplicación se sustenta, como cualquier otra aplicación canónica de clasificación de emociones, en la captura de un vídeo, un extractor de características el cual es el encargado de extraer las características faciales y los datos necesarios para operar correctamente, y finalmente por un clasificador de emociones entrenado y embebido en la aplicación desarrollada.

La Figura 3-1 muestra un diagrama de flujo con el que se tendrá una primera aproximación a la funcionalidad de la aplicación a desarrollar.
De la figura podemos extraer que es necesario un dispositivo de captura de vídeo, cuyo archivo de vídeo generado se le pasa como datos de entrada al extractor de características. Éste se encarga de conseguir datos del vídeo a analizar, que van a parar al clasificador de emociones para, valga la redundancia, detectar y localizar las emociones, que se visualizan en la interfaz gráfica de la aplicación.

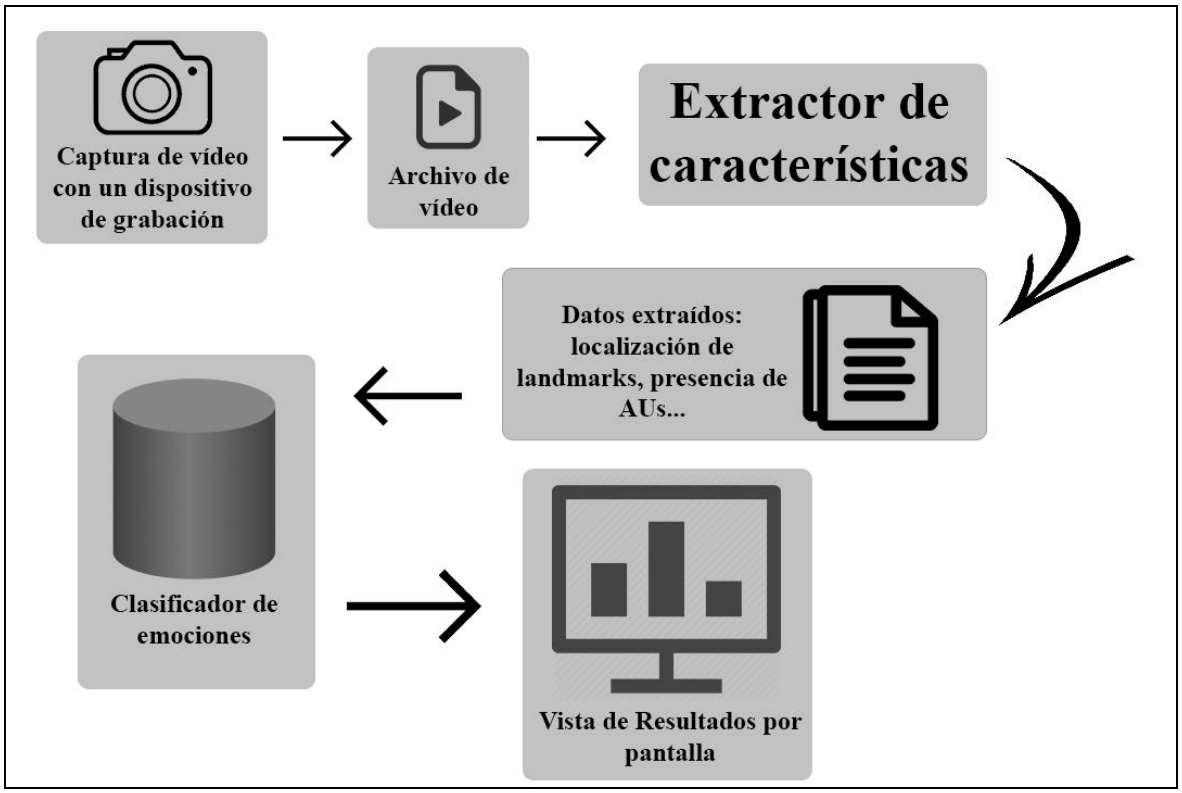

Figura 3-1 Diagrama funcional.

3.2 Descripción de los módulos

En esta sección se describen los distintos módulos mencionados y que se pueden apreciar en la figura, así como también se explica el porqué de cada módulo y el proceso seguido para conseguir el sistema final. Se parte de la descripción del extractor de características, ya que, de la decisión tomada para su desarrollo, depende en gran parte el resto del sistema (formato de entrada y características de entrada al clasificador),

3.2.1 Extractor de características: *OpenFace*

El extractor de características es uno de los módulos más relevantes de este sistema, pues de su eficacia depende el buen funcionamiento de la aplicación diseñada, de ahí que se vaya a explicar lo primero. En este caso se ha escogido utilizar la herramienta *OpenFace* [23].

OpenFace es una herramienta diseñada originalmente por Tadas Baltrušaitis en colaboración con el laboratorio *CMU MultiComp* liderado por el Profesor Louis – Philippe Morency. Se trata de una herramienta de código abierto aún bajo desarrollo por el diseñador original y por otros investigadores como Erroll Wood, Amir Zadeh y Yao Chong Lim. [https://github.com/TadasBaltrusaitis/OpenFace/wiki]

A continuación, se relatará con detalle cada una de las tecnologías presentes en esta herramienta, siendo éstas la detección de *landmarks* faciales, la estimación de la pose de la cabeza, el seguimiento de la mirada y el detector de *Action Units* faciales.

3.2.1.1 Detección y Seguimiento de Landmarks Faciales

La herramienta *OpenFace* usa CLNF [24] (*Conditional Local Neural Fields*), que es una instancia de los CLM [25] (*Constrained Local Models*), para el seguimiento y detección de los *landmarks* faciales.

CLNF se sostiene sobre dos componentes, un modelo de distribución de puntos el cual es el encargado de capturar variaciones de forma de los *landmarks* y otro que captura las variaciones locales de apariencia de cada *landmark*. En este caso, el CLNF está llevado un paso más hacia delante ya que no se localizan los 68 *landmarks* faciales de igual manera, sino que se decidió separar el entrenamiento en tres, haciendo así que el modelo reconozca de manera separada los ojos, labios y por otro lado la zona de las cejas.

Para esta labor, se utiliza un paso previo de localización de caras que se llama "*Face Validation*" que implica que el modelo, formado por una red neuronal convolucional y entrenado con dos bases de datos distintas, tiene que ser reinicializado. En caso de entornos reales se inicializa el modelo y se proponen varias hipótesis para la localización de *landmarks*, escogiéndose definitivamente la hipótesis que más converja con los modelos entrenados, haciendo así que sea fiable esta inicialización, pero con el coste de que se ralentiza esta puesta en marcha.

Cabe mencionar por último que la herramienta es capaz de distinguir y hacer el seguimiento de múltiples caras gracias a un módulo lógico capaz de captar cuando una cara sigue en el cuadro y cuando se va del plano.

3.2.1.2 Estimación de la Pose de la Cabeza

A pesar de que esta tecnología no afecta en ningún sentido al sistema de reconocimiento de emociones desarrollado, no está de más mencionarlo. Simplemente, mencionar que *OpenFace* es capaz de detectar la pose de la cabeza gracias a que dentro del CLNF anteriormente explicado se utiliza una representación en 3D de *landmarks* faciales que son proyectados en la imagen usando una proyección ortográfica de la cámara.
Para dar cierre a este punto, resaltamos que para que esta herramienta sea capaz de estimar la posición de la cabeza de manera precisa se le han de pasar parámetros de calibración de la cámara como bien pueden ser la distancia focal de la cámara, de otro modo se trabaja con una estimación "*best effort*" basada en el tamaño de la imagen.

3.2.1.3 Estimación y Seguimiento de la Mirada

Este es otro punto que, a pesar de que no se emplee en la aplicación final diseñada, está bien revisar y explicar las posibilidades que se ofrecen al contar con esta tecnología.
La herramienta *OpenFace* es capaz de detectar el iris, la pupila e incluso los párpados gracias también al CLNF empleado en la detección o estimación usado en los anteriores puntos. Para entrenar este CLNF se usó una base de datos llamada "*SynthesEyes*" [26] y el modelo resultante igualaba e incluso mejoraba por muy poco los resultados de los métodos de estado del arte hasta el momento, lo cual probó su eficacia. La manera con la que se consigue estimar la mirada es el siguiente: primero se consigue localizar el ojo y la pupila gracias al modelo CLNF, consiguiendo de esta manera información para poder extraer los vectores directamente relacionados con la dirección de la mirada individualmente para cada ojo; después y a través de "disparar" lo que denominan un "rayo" desde el centro de la toma hacia la pupila se consigue hacer una localización 3D de la pupila, resultando en un vector desde el centro de un modelo 3D de un globo ocular hasta las coordenadas resultantes de la pupila, que termina por ser la dirección estimada de la mirada.

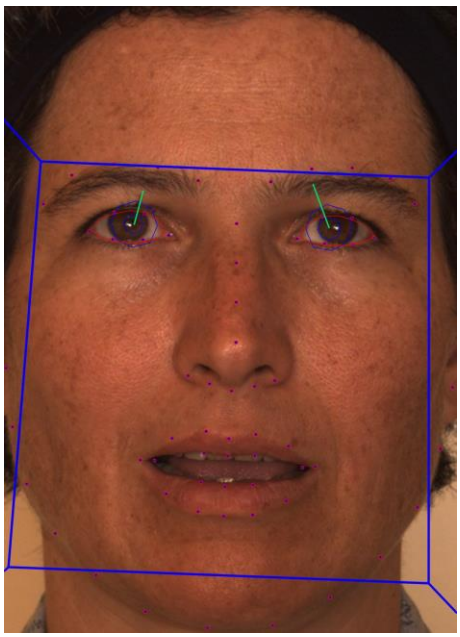

Figura 3-2 Características extraídas por *OpenFace*

En la Figura 3-2 podemos ver el resultado de un análisis real realizado por *OpenFaceOffline.exe* a una imagen estática con el fin de poder ver cada punto explicado de esta herramienta hasta el momento. Se puede apreciar cada una de las funcionalidades explicadas hasta ahora, la presencia de los *landmarks* faciales, representados mediante los puntos de color azul y rojo en zonas como los labios, nariz y cejas; la estimación de la pose de la cabeza que está mostrada mediante la aparición de un cuadrado azul bordeando la cara; y finalmente la estimación de la mirada, en la que podemos apreciar que se ha delineado la zona de los ojos con una línea roja, se ha aproximado la zona del iris y la pupila con dos circunferencias azules y se ha representado el vector que representa la dirección de la mirada con una línea verde con origen en la estimación de la localización de la pupila.

3.2.1.4 Detección de Action Units

Por último, y siendo precisamente el pilar más importante de todos, llegamos a la tecnología que en el sistema desarrollado nos es imprescindible. El módulo de detección de presencia y estimación de intensidad de las *Action Units* se basa en otros recientes *frameworks* cuya finalidad es detectar AUs, mejorando el rendimiento en esta tarea para flujos de vídeo en entornos no controlados en iluminación o ruido, lo cual no estaba del todo conseguido en los *frameworks* en los que se basa.

En el apartado de Estado del Arte se mencionó la existencia de un total de 42 *Action Units* pero sin embargo el *framework OpenFace* es capaz de detectar 18 de ellos, siendo estos los nombrados en la siguiente figura.

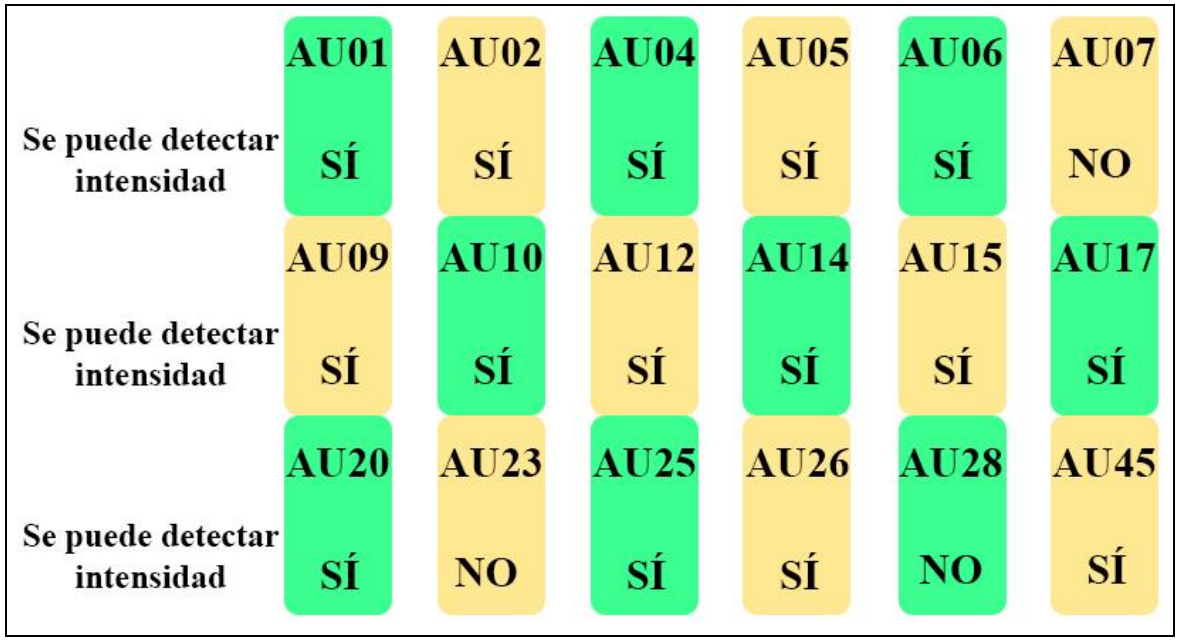

Figura 3-3 *Action Units* detectables por *OpenFace*

Tal y como se ha mencionado, este módulo no solamente detecta la presencia de *AUs* o en su defecto su ausencia, sino que además es capaz de estimar la intensidad de la *Action Unit* en presencia de ella en un rango continuo de 0 a 5. Esta estimación no es trivial, puesto que el ser humano en relaciones humanas en la vida real no es muy expresivo habitualmente, por ende, saber hacer un baremo de intensidades preciso es una tarea compleja.

De todos modos, *OpenFace* incluye una corrección que ofrece resultados mejores a los del Estado del Arte anteriores, siendo esta la sustracción a la predicción del percentil n-ésimo de la predicción de intensidad dado que este arreglo ajusta más los valores de intensidad que otras herramientas que suelen por lo general sobreestimar o subestimar el valor de las intensidades de las *Action Units*.

Para conseguir la predicción de presencia de las *AUs* se emplea un *kernel* lineal SVM y para la intensidad se utiliza un *kernel* lineal SVR; usándose como características de entrada a estos *kernels* una concatenación de HOGs de dimensión reducida y características faciales de forma extraídos gracias al CLNF.

Cabe destacar el hecho de que el reconocimiento de ciertas *Action Units* no es tan fiable como el de otras debido a la falta de representación, según constatan en el artículo, de ciertas *AUs*, es decir, por la aparición más constante de unas que de otras en las bases de datos con las cuales se ha entrenado al sistema. Este problema es solucionable con el paso del tiempo y el incremento de bases de datos disponibles.

Sin embargo, y para finalizar este apartado, es relevante constatar que el porcentaje de acierto de esta compleja tarea se sitúa en un 56% para vídeos (sucesiones de imágenes) y en un 43% para imágenes únicas, siendo porcentajes bastante altos en este campo.

3.2.1.5 ¿Por qué OpenFace?

Ahora que ya conocemos en profundidad *OpenFace* y todo aquello que nos ofrece, exploraremos el porqué de la elección de esta herramienta.
Como comienzo, creo que es importante resaltar el hecho de que se trata de una herramienta de código abierto y de uso público, lo cual es de agradecer ya que con esto se fomenta el interés por la materia, así como se promueve el uso y desarrollo de software libre.

OpenFace no solo ofrece el código de desarrollador para que se desarrolle y compile en la computadora que se quiera usar, sino que además de ello posee varios ejecutables por separado completamente funcionales para poder trabajar con ellos y explorar sus posibilidades.

Se ofrecen un total de tres ejecutables, siendo estos *OpenFaceOffline.exe*, habiéndose mencionado este primero con anterioridad, *HeadPoseLive.exe* y *OpenFaceDemo.exe*.
El primero de los tres listados, *OpenFaceOffline.exe*, es el más importante en este proyecto debido a que es el ejecutable que se usa para extraer las características del vídeo y engloba las funcionalidades de los otros dos, por consiguiente, se explicará qué es capaz de ofrecer cada uno de ellos.

- *HeadPoseLive.exe*: Se trata de un ejecutable cuya función es la de analizar la posición de la cabeza. En la Figura 3-4 podemos ver el aspecto que tiene la aplicación en pleno funcionamiento, donde podemos ver por pantalla los resultados que extrae.

Una vez terminado el análisis, la aplicación devuelve un fichero de texto en el que en cada línea se lista el *timestamp*, la pose X, Y y Z en milímetros y los ángulos de giro de la cabeza en esos tres ejes.

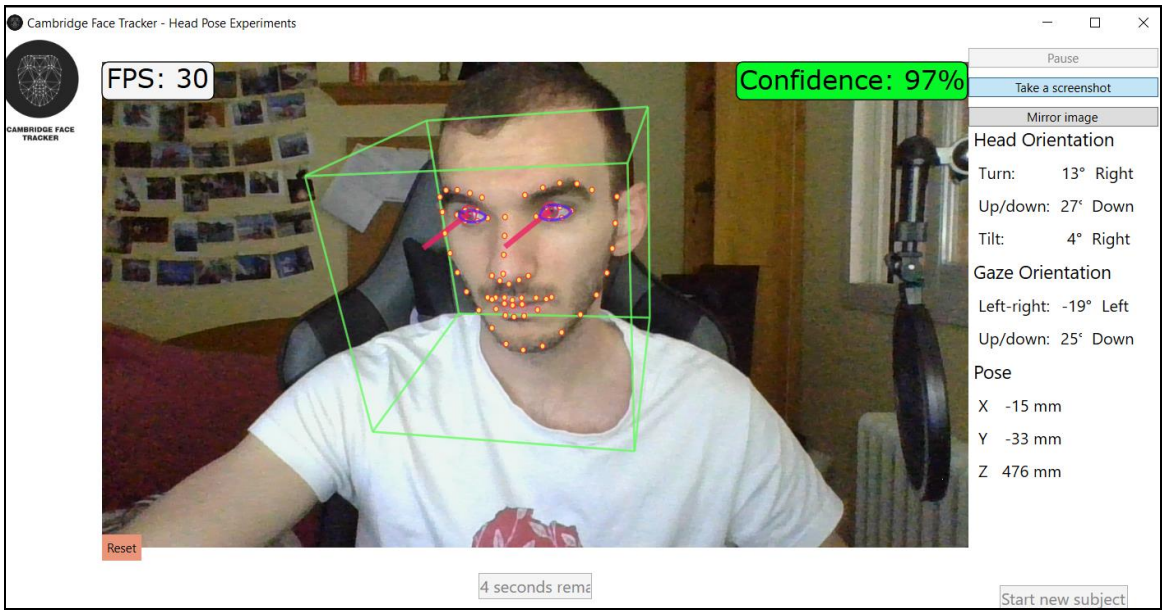

Figura 3-4 *Aplicación HeadPoseLive.exe*

- *OpenFaceDemo.exe*: Se trata de otro ejecutable cuya función es mostrar por pantalla otros resultados que podrían ser de interés para un análisis de comportamiento mediante vídeo, ya que recoge y muestra datos referidos a la intensidad de la sonrisa o con la que se frunce el ceño, la intensidad de elevación o descenso de las cejas o la apertura de los ojos y el arrugamiento de la zona de la nariz.
  También incluye una estimación de la mirada y la estimación de la pose de la cabeza, absorbiendo por lo tanto la funcionalidad entera del ejecutable descrito anteriormente.

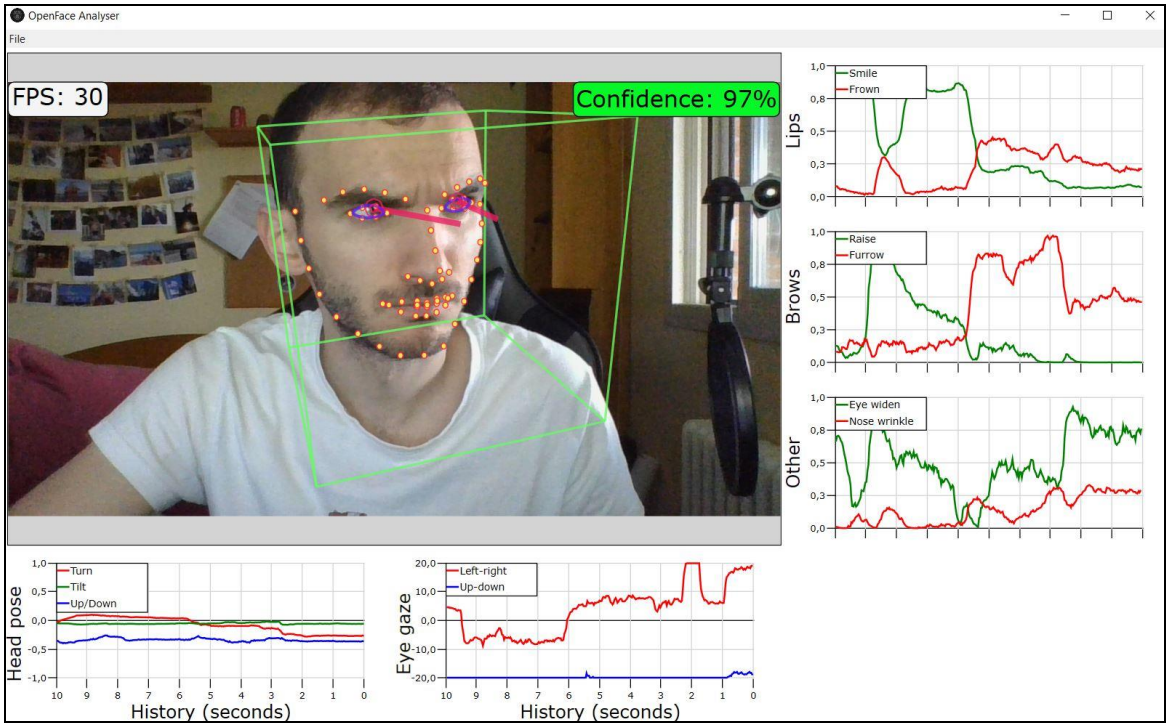

Figura 3-5 Aplicación *OpenFaceDemo.exe*

Este ejecutable difiere del resto en que no devuelve ningún tipo de documento con características extraídas, simplemente es una representación visual del fichero de vídeo analizado. En la Figura 3-5 se puede apreciar todo lo descrito de esta aplicación.

- *OpenFaceOffline.exe*: Este es el ejecutable más relevante de los tres, no solo por ser aquel que se usa en este proyecto, sino además porque incluye las funcionalidades de los otros dos y además incluye el detector de *Action Units* y su intensidad. Además, contiene elementos de visualización en los que se puede ver únicamente la zona de la cara recortada y los Histogramas de Gradientes Orientados (HOG). En la Figura 3-6 se puede visualizar las características descritas.

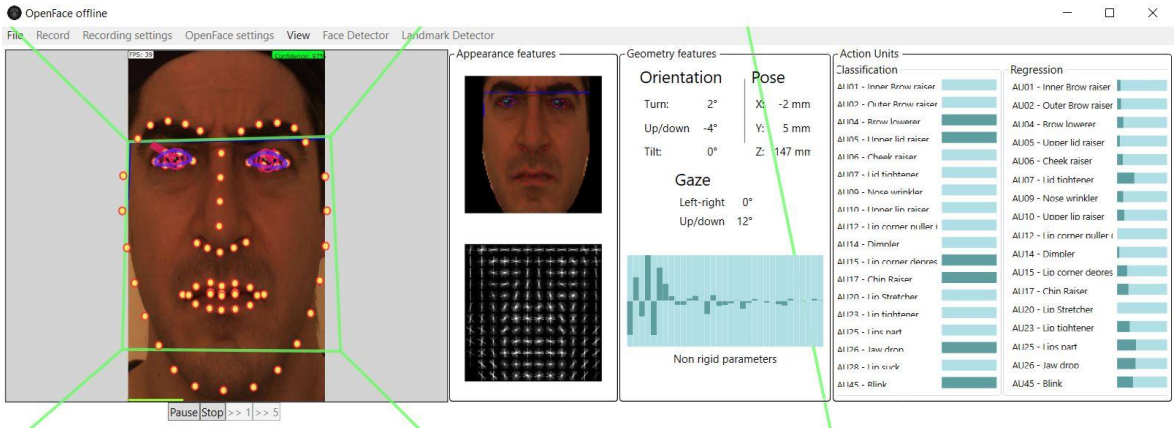

Figura 3-6 Aplicación *OpenFaceOffline.exe*

En este caso sí que contamos con que el resultado del análisis se devuelve en un fichero, siendo este fichero un documento de valores separado por comas (.csv) en el que cada fila

corresponde a un fotograma analizado del vídeo. En estas filas se incluyen los datos de posición de los *landmarks*, la pose de la cabeza, pero sobre todo y más importante, la aparición de las *Action Units* y sus correspondientes intensidades. De ahí se extraen los datos que interesan, que son el número de fotograma y todo lo relacionado con las *Action Units*, siendo estos datos los necesarios para la entrada del clasificador.

Ahora que ya está visto todo lo que puede ofrecer, la decisión de optar por usar *OpenFace*, más concretamente de usar *OpenFaceOffline.exe*, fue clara desde el comienzo del proyecto, puesto a que bien se podría haber optado a hacer un sistema similar a este desde cero, preferimos usar uno que ofrecía buenos resultados y estaba optimizado desde el principio, más aun siendo software libre y teniendo en cuenta de que, a pesar de que no se tenía a otro candidato más que *OpenFace*, muchos software que puedan ofrecer funcionalidades similares a las que se ofrecen tienen licencia de pago y preferimos apostar por software de código libre.

3.2.2 Captura y formato de vídeo

El primer paso a tomar a la hora de querer usar este sistema es capturar un vídeo de una cara para su posterior procesamiento. Para ello, no es necesario disponer de ningún material específico, el sistema es en este sentido bastante robusto pues es capaz de procesar flujos de vídeo tomados con cualquier tipo de dispositivo de captura, ya sea una cámara profesional, una cámara web o la cámara de un smartphone.

Los vídeos capturados pueden estar en horizontal o en vertical, la disposición no afectará a los resultados finales. El problema entonces recae en el cambio horizontal a vertical o viceversa durante el vídeo. En ese caso sí que existirán dificultades de procesamiento, pudiendo dar lugar a numerosos errores.

Por otro lado, la continuidad del vídeo es otro elemento en el que esta herramienta es bastante robusta dado a que no importa si existen saltos temporales o cortes dentro del vídeo. Mientras exista una cara evidente en el vídeo, el sistema funcionará sin problemas.

Finalmente, el formato en el que se capture el vídeo original tiene que ser uno de los aceptados por el software *OpenFaceOffline.exe* de *OpenFace* [23], que son los formatos .avi, .webm, .wmv, .mov, .mpg, .mpeg y .mp4. Cualquier otra extensión que tenga el fichero de vídeo de origen no será aceptada, por tanto, se aconseja grabar y codificar en uno de los formatos requeridos o bien convertir el fichero no aceptado a uno con el que sí se pueda trabajar.

3.2.3 Clasificador

Finalmente, para que el sistema diseñado funcione correctamente, se ha de tener un clasificador de emociones.

Se cuenta con un clasificador entrenado en el *framework* de MatLab. Este clasificador debería ser capaz de distinguir y clasificar entre seis emociones, que son Enfado, Miedo, Repulsión, Felicidad, Tristeza y Sorpresa.

Estas son las seis emociones más básicas, de ahí que se haya decidido, no solo en este proyecto sino en la mayoría de proyectos a nivel mundial referidos al campo de la detección y clasificación de emociones, trabajar única y exclusivamente con estas y no incluir otras que, en principio, son mucho más complejas e incluso son combinaciones entre las anteriormente mencionadas.

Como datos de entrada al clasificador hay que pasarle los datos recogidos únicamente de las *Action Units*, es decir, hay que pasarle los datos de detección o no de cada *AU* y la intensidad de cada una, siendo de intensidad nula si no se ha detectado presencia de *AU* o en el rango de 0 a 5 en caso de que sí se detectara presencia.

Destacar sobre todo la complejidad del problema, no porque nos encontramos en un caso en el que la clasificación se hace bajo un subespacio de 35 dimensiones, sino por el hecho de que las emociones a clasificar comparten mucha similitud, siendo compleja la separabilidad.

3.2.3.1 Primera aproximación del Clasificador

Como inicio en este proyecto, tras consultar y estudiar artículos relacionados con el tema en cuestión, se optó por simplificar el problema.

Todos los sistemas que trabajaban en el reconocimiento y detección de emociones se basaban en el desarrollo de un clasificador por medio de SVMs y debido a esto primero se planteó un cambio en cuanto al clasificador o detector, tomando una aproximación heurística.

Contrastando la información obtenida de distintos artículos y sitios web como puede ser la web "IMotions" (https://imotions.com/blog/facial-action-coding-system/ -ultimo acceso verificado junio 2020-), se propuso probar si un sistema sencillo basado en aparición de *Action Units* en el que mediante la combinación de ellas en un mismo fotograma pudiera dar como resultado la detección de una emoción.

Dicho razonamiento puede verse representado en la siguiente imagen.

Happiness / Joy	6 + 12	Cheek Raiser, Lip Corner Puller	

Figura 3-7 (extraída de https://imotions.com/blog/facial-action-coding-system/)Felicidad como combinación de presencia de *AUs*

Como se puede apreciar en la imagen, la idea está fundamentada, sin embargo, se trata de algo mucho más complicado.

Las combinaciones propuestas, tras analizar la información disponible, para que esto diera resultado eran las siguientes:

- Felicidad: AU_06 + AU_12.
- Tristeza: AU_01 + AU_04 + AU_15.
- Sorpresa: AU_01 + AU_02 + AU_05 + AU_26.
- Miedo: AU_01 + AU_02 + AU_04 + AU_05 + AU_07 + AU_20 + AU_26.
- Enfado: AU_04 + AU_05 + AU_07 + AU_23.
- Repulsión: AU_09 + AU_15 + AU_16.

Viendo estas combinaciones, se puede inferir que la lógica a implementar para realizar el detector es bastante simple. Sin embargo, ya nos encontrábamos un problema que, aunque menor, ya indicaba que esta aproximación no sería la adecuada, y ese problema es el de que se necesita una *Action Unit* (AU_16) de la que no podemos disponer, debido a que *OpenFace* no está entrenado para detectar esta *Action Unit*, para reconocer la emoción Repulsión.

Por otro lado, cabe mencionar el hecho de que estamos dejando de lado la intensidad de cada *AU*, decantándonos por dar relevancia única y exclusivamente a la presencia o no, lo cual nos lleva a otro problema: el hecho de que cierta combinación de *AUs* de como resultado una emoción no implica que no exista ninguna otra *Action Unit* más. Es decir, en una imagen etiquetada como que representa "Enfado" no solamente se consiguen extraer los *AUs* que conforman esa emoción según esta lógica, sino que además se pueden extraer otras con una intensidad ínfima que puedan dar lugar a error.

Efectivamente, la lógica inicial implementada para la detección de estas seis emociones era muy simple: identificar si existían o no las *Action Units* requeridas para cada emoción siendo el resultado de emoción detectada sí y sólo si se detectaban todas y cada una de las *Action Units* que se necesitaban. Los resultados que se arrojaban no fueron buenos, no superando cuatro de las seis emociones el 15% de acierto, una de las dos restantes con un porcentaje del 25% y sorprendentemente luego teníamos la emoción Felicidad con un 98% de acierto. Este último porcentaje tan alto se debe a que tal y como se ha relatado anteriormente, la emoción Felicidad necesitaba únicamente de dos *Action Units* para ser detectadas y que estas dos eran únicas de esta emoción, en ninguna otra se contaba con estas *AUs* para la detección, por tanto, era una emoción segregada del resto al no compartir características.

Sin contar con esta emoción, el resto tenían porcentajes de acierto muy por debajo de lo deseable, y no solamente eso, sino que además tenían un porcentaje de error muy alto debido a que en un solo fotograma se clasificaba la emoción detectada no de manera única, sino que tal clasificador detectaba diversas emociones a la vez, siendo tal cosa inconcebible. Esto era posible gracias a que la lógica que se propuso no contaba la intensidad de las *Action Units* y a que las combinaciones de *AUs* propuestas no eran exclusivas, se compartían *AUs* en todas las emociones salvo en Felicidad como ya se ha mencionado.

Para intentar solventar esta problemática, se optó por tener en cuenta las intensidades, estableciendo ciertos límites mínimos para dar como válida la presencia o no de una *Action Unit*.

Los valores de estos límites fueron fijados mediante un estudio riguroso de los valores de intensidad que arrojaba *OpenFace* sobre los vídeos de *test*. Se fijaron esos límites en la media ponderada de los valores de intensidad de cada *Action Unit* que intervenía en la detección de cada emoción.
Esta decisión no hizo más que agrandar el problema, porque si bien es cierto que se solucionó parcialmente el tema de la detección de varias emociones para un solo fotograma, el porcentaje de acierto cayó más de un 4% en todas las emociones, poniéndose de manifiesto que esta aproximación que en un principio se decidió llevar a cabo para simplificar el problema no había sino agrandado la problemática, por lo que se decidió desechar todo lo avanzado en esta propuesta para reorientar el trabajo.

Finalmente, se decidió seguir con la moda actual y entrenar un clasificador que se explicará más adelante tras comprobar que la aproximación heurística no era una solución válida, excepto para el caso de detectar la emoción de felicidad, caso en el cual podríamos optar por la opción de elaborar un detector de la manera propuesta para disminuir significativamente la complejidad computacional.

3.2.4 Visualización

Este apartado tratará sobre la visualización de los resultados finales del proceso, es decir, se enseñará cómo funciona la aplicación para poder ver los resultados finales por pantalla.

Los resultados finales se van mostrando por pantalla a medida que se hace la clasificación de la emoción de cada fotograma analizado, es decir, no se trabaja primero la clasificación para luego únicamente mostrar los resultados.

En la Figura 3-8 podemos visualizar como se ve la aplicación diseñada para mostrar los resultados por pantalla antes de inicializar el sistema.

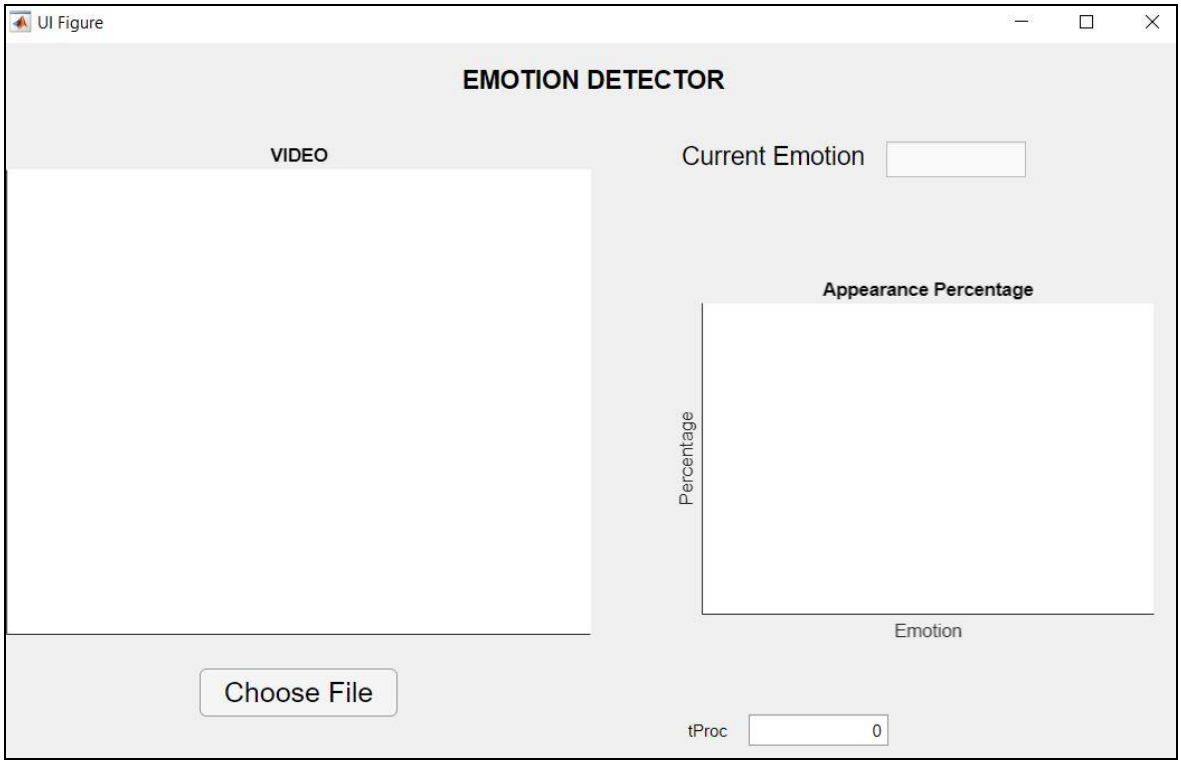

Figura 3-8 Aplicación desarrollada para la visualización

Se trata de una aplicación diseñada en el *App Designer* de MatLab. Tal y como se puede apreciar, se reserva espacio en la pantalla para poder visualizar el vídeo que se ha analizado a la izquierda de la aplicación, en donde no aparecerá el vídeo de origen, sino aquel que ya está analizado por *OpenFace*, contando este con los *landmarks* faciales, la estimación de la mirada y la estimación de la posición de la cabeza.

En la parte de la derecha se puede observar que se reserva un espacio de tamaño pequeño en la parte superior donde, por cada fotograma, aparecerá la etiqueta de la emoción que se ha detectado.

Inmediatamente debajo de la sección donde irá la etiqueta de la emoción detectada, tenemos un espacio reservado para que se elabore un diagrama de barras dinámico, lo cual significa que está en constante cambio mientras la clasificación se esté llevando a cabo, donde se podrá visualizar el porcentaje de aparición de cada una de las seis emociones capaces de detectar a lo largo de la clasificación de la secuencia.

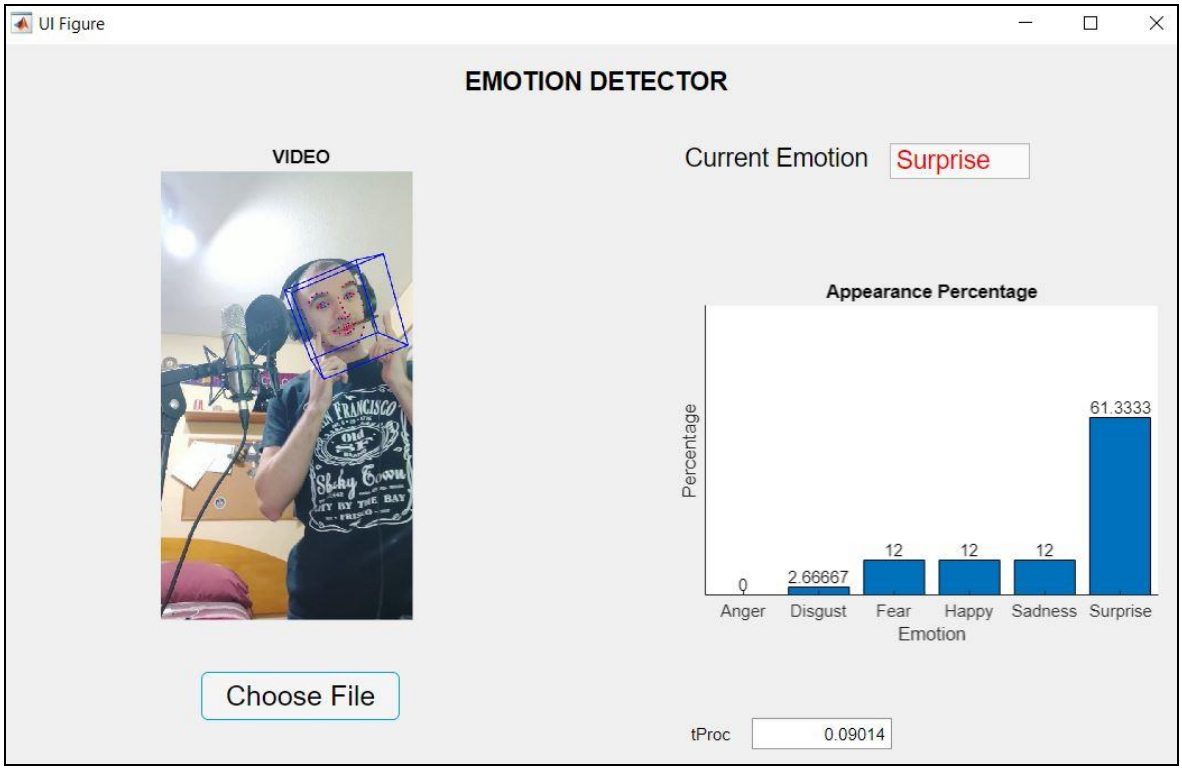

Figura 3-9 Aplicación desarrollada en funcionamiento

En la Figura 3-9 se puede apreciar una simulación de la aplicación en funcionamiento con un vídeo real, grabado en vertical. Cabe recordar que el sistema entero funciona tanto con vídeos en vertical, como se puede apreciar, como con vídeos en horizontal.

El cuadro blanco que aparecía en la Figura 3-8 que reservaba el espacio para el vídeo vemos que desaparece, mostrando únicamente el vídeo analizado por *OpenFaceOffline* sin ningún recuadro o bordes blancos que lo enmarquen, quedando de esta manera más limpia la visualización.

A la derecha se puede comprobar que el recuadro pequeño reservado para la etiqueta de la emoción correspondiente al fotograma actual se encuentra relleno con tal cosa, siendo esta en particular la emoción "*Surprise*" (Sorpresa). Las etiquetas de las emociones mostradas en ese recuadro están con un color de fuente llamativo para que la visualización sea sencilla y llamativa.

Por último, inmediatamente por debajo de la sección donde se encuentra la etiqueta de la emoción detectada tenemos el cuadro de porcentajes rellenándose y actualizándose de manera dinámica a medida que avanza la clasificación. Es preciso recalcar que se trata de un porcentaje de aparición, no del número de fotogramas en el que tenemos una emoción concreta, lo cual facilita la visualización y el análisis final del vídeo.
Estos porcentajes tienen una resolución de cinco cifras decimales, por tanto, para vídeos muy largos, aunque también para los cortos claro está, obtenemos una precisión bastante alta del número de apariciones por emoción.

4 Integración, pruebas y resultados

El propósito de esta sección que ahora nos ocupa es el de mostrar el proceso llevado a cabo para realizar el clasificador definitivo de este proyecto. También se profundizará en los resultados que ofrece el clasificador. Asimismo, se ahondará en el código de la aplicación.

4.1 Datasets para el entrenamiento

Esta sección tratará el tema relacionado con las bases de datos que se buscaron para llevar a cabo este proyecto, la selección de aquella que mejor se ajustaba a las necesidades de éste y el porqué de las decisiones tomadas sobre este tema.

La necesidad principal en este proyecto era la de encontrar un clasificador que operase adecuadamente con un porcentaje de acierto similar a lo comparable con el estado del arte. Para ello, y visto ya el desarrollo que se llevó con las aproximaciones a un clasificador eficaz, era preciso contar con una base de datos de imágenes con la que poder entrenar nuestro clasificador a desarrollar.

Antes de comenzar con el proceso de búsqueda de las bases de datos, se tomó la decisión de consultar sobre este tema al doctor Tadas Baltrušaitis debido a que se trata de una persona con amplia experiencia en el campo de reconocimiento facial y de detección de *Action Units*, por lo que su guía podía ser de gran ayuda.

Al contactar con él, aparte de manifestar su gratitud por la muestra de interés en el software de *OpenFace*, nos redujo las posibilidades de escoger entre muchas bases de datos distintas a un sencillo conjunto de dos bases de datos, habiendo sido ambas usadas por él mismo y su equipo para desarrollar su software.

Estas dos bases de datos propuestas por el Dr. Baltrušaitis son las a continuación mencionadas.

4.1.1 The Extended Cohn-Kanade Dataset

Esta base de datos, conocida también como CK+ [27], es un conjunto de imágenes tomadas entre personas del rango de 18 a 50 años de edad, tomadas con dos cámaras iguales Panasonic AG-7500 bajo una iluminación controlada.
A los participantes en la obtención de las imágenes para la base de datos se les indicó realizar un total de disposiciones faciales, las cuales se buscaban que contuvieran una o un conjunto de *Action Units* presentes, de 23.

Además de esto, algunas imágenes se etiquetaron como ciertas emociones, que fueron pedidas expresamente por los realizadores del estudio, sin embargo, no todas las imágenes captadas en este proceso de recolección de emociones fueron aceptadas como *ground truth* debido a la difícil tarea de imitación de ciertas emociones. Incluye una emoción adicional a las 6 básicas, siendo esta añadida la emoción de "*Contempt*" (Desprecio).

Un total de 327 sobre las 593 imágenes iniciales pasaron la aprobación de la inspección de los investigadores, pues concluyeron que este grupo de imágenes representaba con certeza lo que se exigía.

Todas las imágenes digitalizadas tienen una dimensión de 640x490 píxeles o bien de 640x480 píxeles, a escala de grises de 8 bits o bien con una profundidad de color de 24 bits.

4.1.2 The Bosphorus Dataset

Esta base de datos [28] consta de un total de 4652 imágenes de un grupo de 105 participantes en el estudio. cuya edad está comprendida mayoritariamente en la franja de los 25 – 35 años. Entre estos participantes, aproximadamente una cuarta parte de ellos son actores o actrices de profesión, figurando como un total de 27.
Las imágenes tomadas son capturas en tres dimensiones obtenidas con un dispositivo digitalizador Inspeck Mega Capturor II 3D con una iluminación controlada. En esta base de datos se dispone por lo tanto de digitalizaciones 3D de la cara, así como de las imágenes en 2D.

A diferencia de la base de datos anteriormente mencionada, los tamaños de las imágenes en este caso no están normalizados, si bien es cierto que todas y cada una de las imágenes 2D que se ofrecen superan los 1000 píxeles de tamaño en cada dimensión.

Por cada sujeto del estudio se proporcionan imágenes correspondientes a la representación de un conjunto de *Action Units* requeridas por los realizadores del estudio, la representación de las 6 emociones básicas en la mayoría de los casos, salvo para determinados sujetos que realizan sólo 5 o incluso 4 de las 6 emociones básicas; y finalmente se proporcionan imágenes de rotación de cara y oclusiones con cabello cubriendo la cara, presencia de gafas y con oclusiones de zonas faciales debido a la presencia de manos.

4.1.3 Decisión final: base de datos escogida para el proyecto

Ahora que ya se ha podido tener una explicación de ambos *datasets*, podemos concretar que se tomó la decisión de avanzar en el proyecto con el *Bosphorus Dataset*.
Esta decisión se fundamenta en el hecho de que las etiquetas de las imágenes que representaban emociones son más fiables bajo el criterio que se tomó que las de la primera opción.

Además de la cantidad más grande de imágenes que posee *Bosphorus Database* cuenta con Imágenes en 3D que a pesar de que no se utilizan para nada en este proyecto podría darse el caso de que se usen en trabajos futuros.

Figura 4-1 Imágenes 2D (Izquierda) y 3D (Derecha) de Bosphorus Database

4.2 Clasificador de Emociones

En el punto 3.2.3 se vio la posibilidad de hacer un clasificador distinto a lo que se hace generalmente, partiendo de una aproximación heurística.

Se pudo comprobar que tal acercamiento no era correcto pues no solucionaba ningún problema ni hacía más sencillo el sistema al no arrojar resultados correctos salvo para el caso de la emoción Felicidad, lo cual hace que sea útil en casos en los que se quiera detectar única y exclusivamente esa emoción, pero no para el resto.

Por lo tanto, y en vista de los malos resultados ofrecidos por el prototipo de clasificador heurístico que se llevó a cabo, se tomó la decisión de entrenar un clasificador por medio de MatLab.

Esto fue posible a una herramienta del toolbox de MatLab "*Machine Learning And Deep Learning*", llamada "*Classification Learner*", que permite entrenar un clasificador con el método que se prefiera, como por ejemplo una SVM. Pero para poder entrenar, primero se necesitan datos sobre los que trabajar.

De la base de datos *Bosphorus Dataset* se agruparon, como primer paso, las imágenes etiquetadas de cada emoción diferente, quedando así 6 conjuntos de imágenes pertenecientes a las 6 emociones que se estudian.
De esas imágenes se formó un vídeo para cada emoción, conformando tal vídeo con una imagen por fotograma de cada sujeto en cada emoción.
Cada vídeo se pasó por *OpenFaceOffline* de manera separada, extrayendo de cada imagen las *Action Units* presentes y sus correspondientes intensidades junto con el resto de datos que se extraen.

Entonces, de cada fichero de datos extraídos se salva únicamente las *AUs* junto con sus intensidades y además se añade manualmente una etiqueta como un parámetro más para poder hacer el entrenamiento. Esta etiqueta es un número del 1 al 6 que especifica de qué emoción es cada conjunto de *AUs* e intensidades, siendo el código de numeración:

- 1: ANGER (Enfado)
- 2: DISGUST (Repulsión)
- 3: FEAR (Miedo)
- 4: HAPPY (Felicidad)
- 5: SADNESS (Tristeza)
- 6: SURPRISE (Sorpresa)

Una vez se ha hecho esto con los 6 ficheros de AUs, intensidades y etiquetas, se junta todo en un fichero que será el usado para el entrenamiento. En la Figura 4-2 se puede apreciar un extracto del fichero conjunto.

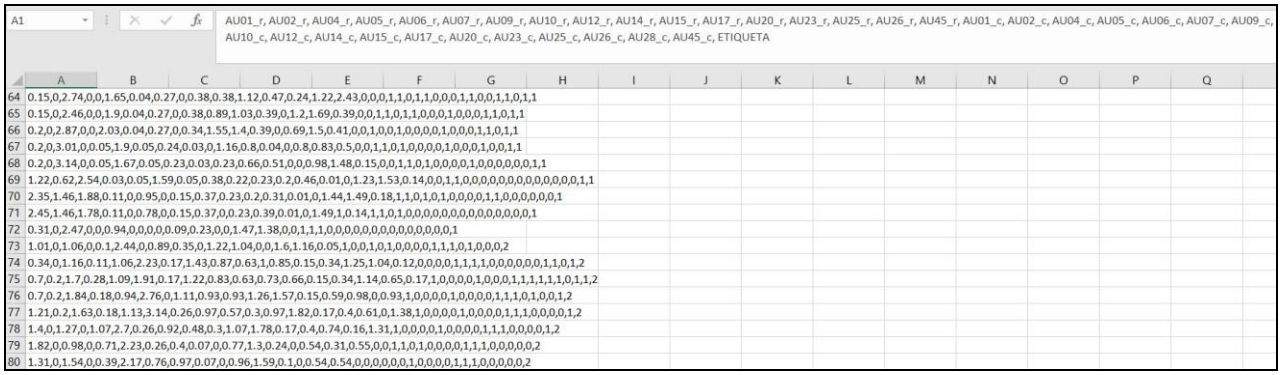

Figura 4-2 Fragmento del fichero de datos extraídos con la etiqueta de emoción correspondiente

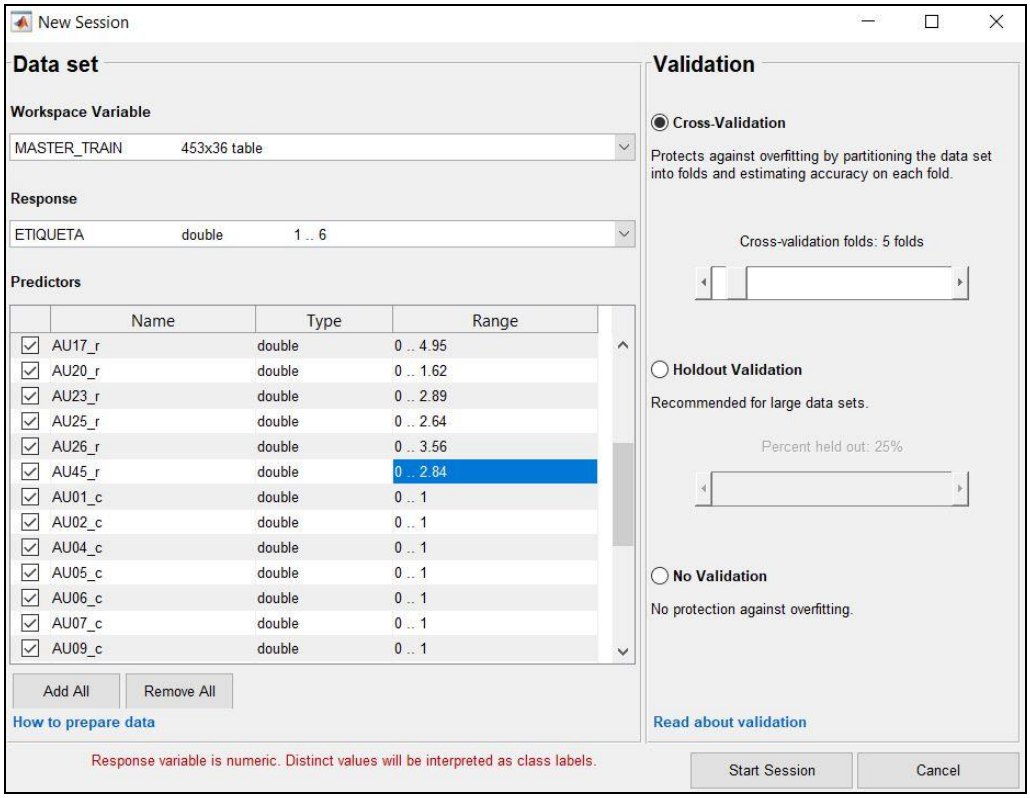

Figura 4-3 Sesión de entrenamiento con "Classification Learner"

En la Figura 4-3 se puede observar la interfaz de la aplicación *Classification Learner* de Matlab para el entrenamiento. "*MASTER_TRAIN*" es el nombre de la tabla que reúne todos los AUs extraídos de todas las imágenes etiquetadas, un total de 453, del *Bosphorus Dataset*.

Cabe destacar el hecho de que no solamente se hizo una sesión de entrenamiento, sino que fueron un total de 10 las sesiones. Este hecho se debe a que en cada sesión se cambiaba el valor de los "*folds*" para realizar una validación cruzada empezando con tamaño 5, aumentándose el tamaño en saltos de 5 y terminando con 50 "*folds"* en la última sesión. La validación cruzada nos sirve para evaluar los resultados y tener garantías de la independencia entre datos de entrenamiento y de test.

En estas 10 sesiones los resultados, aunque similares, variaban en cuanto al porcentaje de acierto de las emociones clasificadas correctamente, dando en cada sesión métodos distintos de clasificación como se verá en breve.

La herramienta que se usa para entrenar el clasificador ofrece muchos métodos distintos de clasificación final tras el entrenamiento, siendo estos:

- Árboles de decisión con 4, 20 o 100 divisiones por rama.
- Clasificadores de Naive-Bayes.
- *Support Vector Machines* (SVM).
- Vecinos más cercanos.
- Conjuntos de clasificadores.

De estos cinco tipos, solo nos interesa recalcar los SVM y los Conjuntos de Clasificadores pues son los modelos escogidos para los distintos números de "*folds*" debido a que fueron aquellos modelos con mayor porcentaje global de acierto.

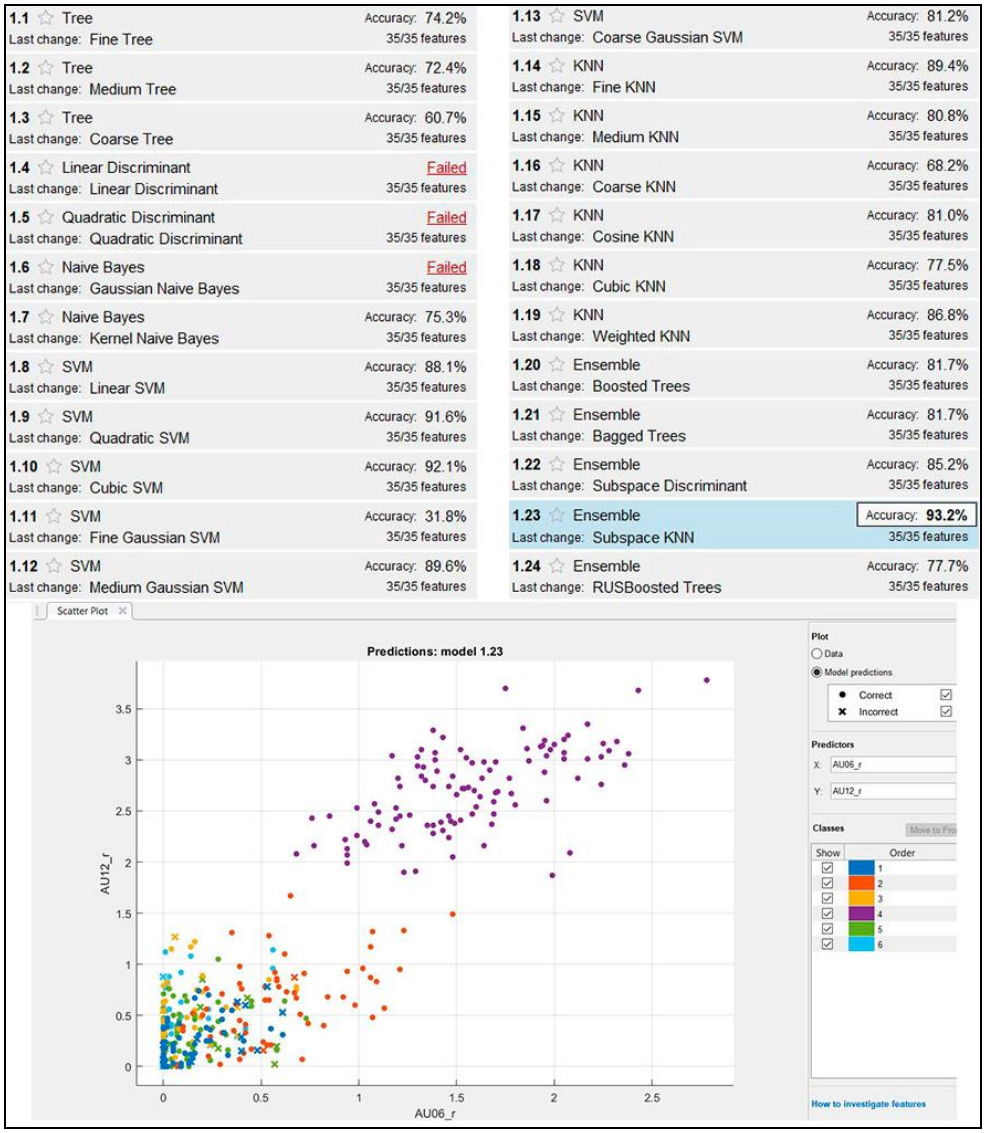

Figura 4-4 Ejemplo de Resultados del "Classification Learner" (20 "folds")

Para las sesiones con conjuntos de validación iguales a 10, 15, 25, 30 y 50 el mejor método de clasificación basándose en porcentaje de acierto es el SVM, mientras que en las sesiones para conjuntos de validación de 5, 20, 35, 40 y 45 el mejor método es el conjunto de clasificadores, concretamente el "Subspace KNN", es decir un "Subspace" con *learners* de vecino más cercano.

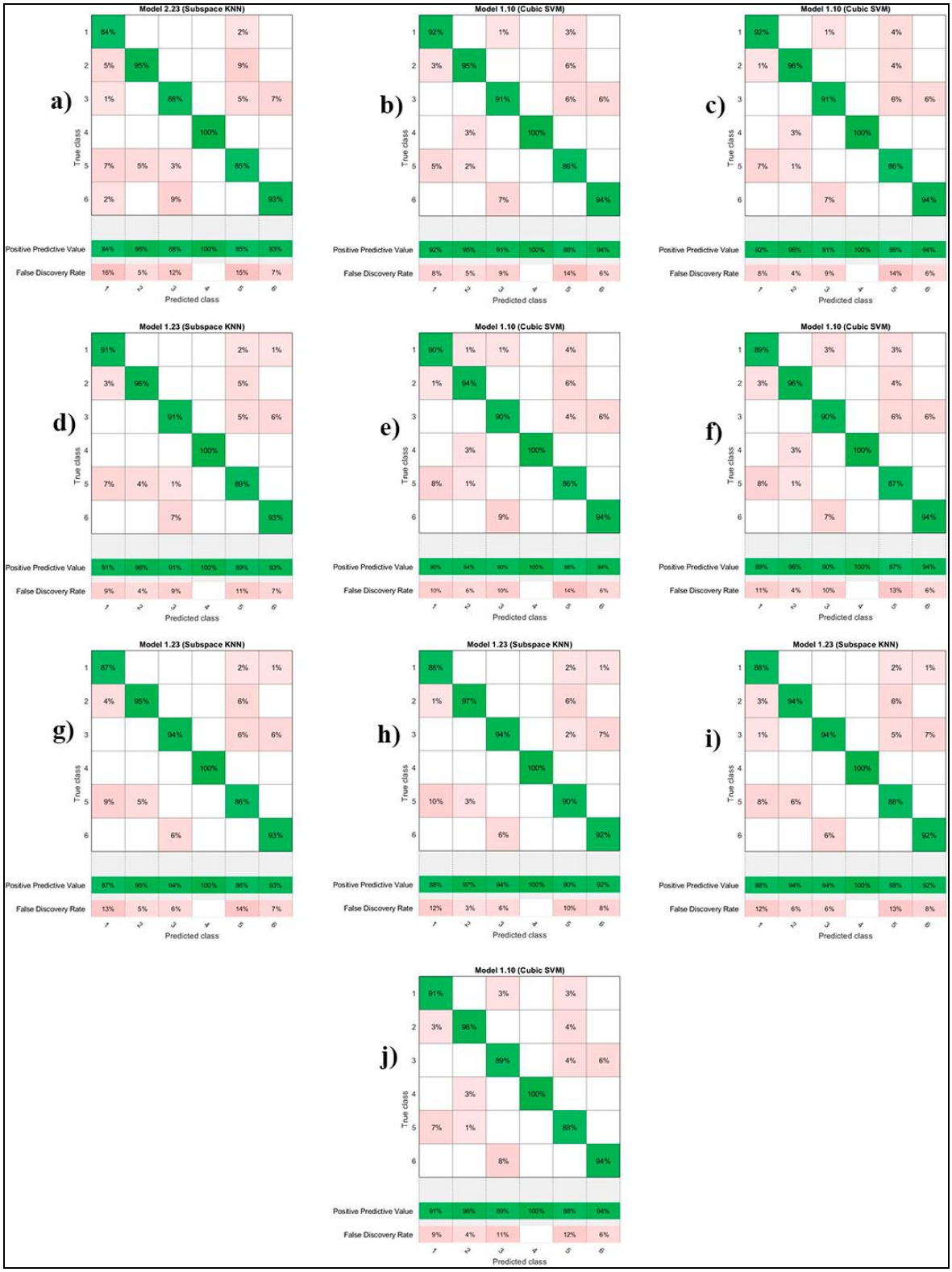

Figura 4-5 Matrices de confusión del mejor clasificador por sesión

En la Figura 4-5 tenemos las matrices de confusión para a) 5 fold cross-validation (Subspace KNN), b) 10 fold cross-validation (Cubic SVM), c) 15 fold cross-validation (Cubic SVM), d) 20 fold cross-validation (Subspace KNN), e) 25 fold cross-validation (Cubic SVM), f) 30 fold cross-validation (Cubic SVM), g) 35 fold cross-validation (Subspace KNN), h) 40 fold cross-validation (Subspace KNN), i) 45 fold cross-validation (Subspace KNN), j) 50 fold cross-validation (Cubic SVM).

Para evaluar los resultados se han estudiados los resultados ofrecidos por el *Classification Learner*, que son:

- Matriz de confusión: tabla de visualización en la que las columnas representan las clases verdaderas y las filas representan las clases predichas. Las predicciones correctas se representan en porcentaje en la diagonal principal, marcada en este caso de color verde.
- *Positive Predictive Value*: valor en porcentaje definido por la siguiente ecuación

$$\frac{n^{o}\ verdaderos\ \text{positivos}}{n^{o}\ verdaderos\ \text{positivos} + n^{o}\ falsos\ \text{positivos}}$$

  El valor ideal para esta métrica es 100%, el peor caso posible 0%.

- *Negative Predictive Value*: valor en porcentaje definido por la siguiente ecuación

$$\frac{n^{o}\ falsos\ \text{positivos}}{n^{o}\ verdaderos\ \text{positivos} + n^{o}\ falsos\ \text{positivos}}$$

  A menor que sea este parámetro, mayor será el clasificador entrenado.

De todas estas sesiones de entrenamiento, se tomó la decisión de mantener dos de los modelos entrenados de los diez totales, que fueron aquellos que dieron como resultado de entrenar el modelo con 10 grupos de validación cruzada y el otro con 20 grupos de validación cruzada.
La decisión de coger estos dos modelos como los posibles candidatos finales está fundamentada en el hecho de que, por lo general y tal y como se comenta en el apartado 2.3.3, lo más común es escoger el modelo SVM con un entrenamiento de validación cruzada con 10 grupos y, por el otro lado, el modelo entrenado con 20 grupos es en el que encontramos un "*Positive Predicted Value*" por encima del 90% en todas las emociones salvo en una de ellas (emoción número 5: Tristeza) cuyo porcentaje se sitúa en un 89%, con lo cual se puede generalizar y decir que todas ellas superan el 90% de "*Positive Predicted Value*".
En la siguiente tabla (Tabla 4-1) podemos comparar ambos modelos y ver sus características a nivel computacional:

TIPO DE CLASIFICADOR	Velocidad de Predicción	Uso de Memoria	Interpretabilidad	Flexibilidad del Modelo
Cubic SVM	Binario: Rápida Multiclase: Lenta	Binario: Medio Multiclase: Grande	Difícil	Media
Subspace KNN	Media	Media	Difícil	Media

Tabla 4-1 Diferencias computacionales entre Cubic SVM y Subspace KNN. Información extraída de [29].

Como se puede comprobar, a nivel computacional es mejor opción optar por el modelo "Subspace KNN" dado que es un problema que es multiclase, ergo a pesar de que la interpretabilidad y la flexibilidad del modelo sean igual de compleja, se puede observar que la velocidad de predicción es más rápida en el segundo caso, así como que se utiliza una memoria menor.

Por tanto y como decisión definitiva, la aplicación desarrollada cuenta con un clasificador "Subspace KNN" entrenado con 20 grupos de validación cruzada.

4.3 Aplicación: Código y Rendimiento

En esta sección se verá el código y se comentarán las limitaciones computacionales de la aplicación desarrollada.

Tras ejecutar la aplicación desde MatLab, lo primero que es necesario hacer es pulsar sobre el botón "Choose File" de la interfaz, en ese momento la aplicación arrancará con lo siguiente.

El código de la aplicación comienza cargando el modelo entrenado escogido, al que se le ha llamado "*subspaceKNN_20foldX_validation.mat*", pidiendo inmediatamente después al usuario que escoja el documento de características extraídas previamente por *OpenFace* y el fichero de vídeo analizado de una ventana emergente, como se puede ver en la siguiente figura.

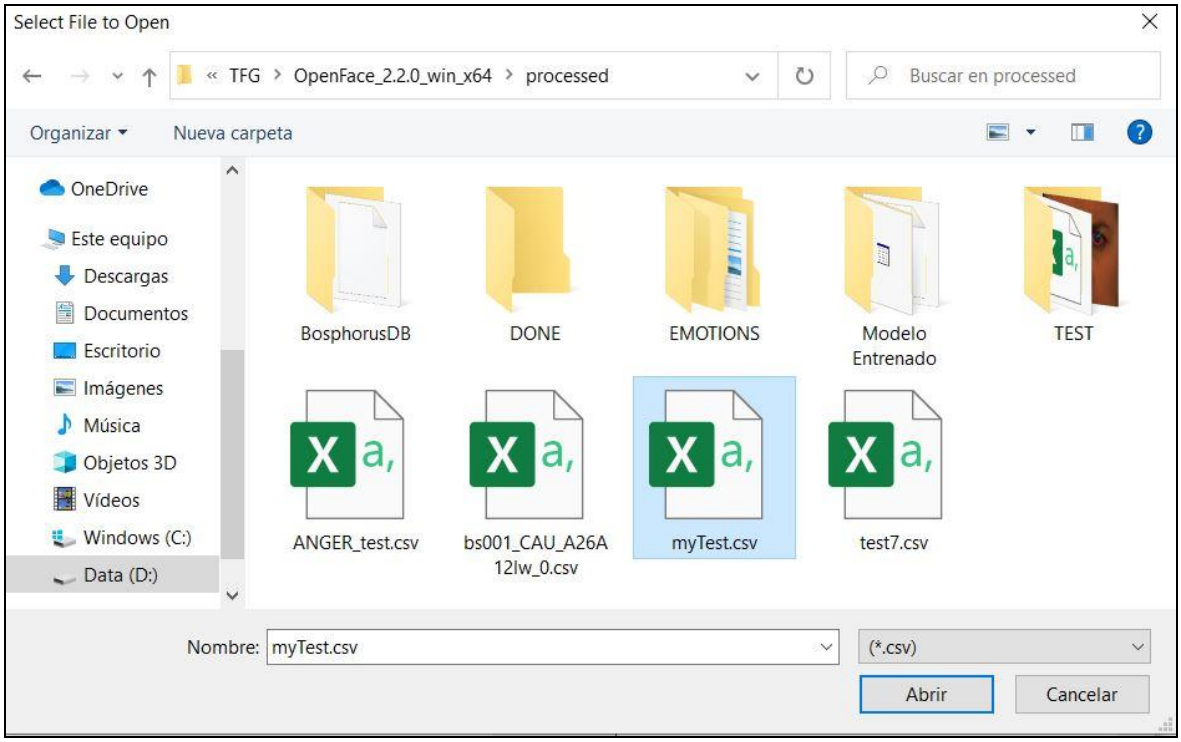

Figura 4-6 Ventana Emergente para la Selección

Después de escoger ambos ficheros, ambos son cargados en la aplicación, el documento de características servirá para poder realizar la clasificación y el fichero de vídeo para la futura visualización.

La lógica siguiente de clasificación y muestra de resultado se realiza fotograma a fotograma, lo cual indica que se realiza la clasificación a medida que avanza el vídeo, no se clasifica previamente.

Código:

```
                tic;
                fila(frame_number,:) = tabla_AUs(frame_number,:);
                prediccion(frame_number) =
subspaceKNN_20foldX_validation.predictFcn(fila(frame_number,:));
                mov(frame_number).cdata = readFrame(v);
                if vidHeight > 720
                    frame_resized =
imresize(mov(frame_number).cdata,0.5);

                elseif vidWidth > 720
                    frame_resized =
imresize(mov(frame_number).cdata,0.5);
                else
                    frame_resized = mov(frame_number).cdata;
                end

                imshow(frame_resized,'Parent',app.UIAxes);

                if  prediccion(frame_number) ==1
                    app.CurrentEmotionEditField.Value = 'Anger';
                    Anger = Anger+1;

                elseif  prediccion(frame_number) ==2
                    app.CurrentEmotionEditField.Value = 'Disgust';
                    Disgust = Disgust+1;

                elseif  prediccion(frame_number) ==3
                    app.CurrentEmotionEditField.Value = 'Fear';
                    Fear = Fear+1;

                elseif  prediccion(frame_number) ==4
                    app.CurrentEmotionEditField.Value = 'Happy';
                    Happy = Happy+1;

                elseif  prediccion(frame_number) ==5
                    app.CurrentEmotionEditField.Value = 'Sadness';
                    Sadness = Sadness+1;

                elseif  prediccion(frame_number) ==6
                    app.CurrentEmotionEditField.Value =
'Surprise';
                    Surprise = Surprise+1;
                end
```

```
                y = [(Anger*100)/frame_number
(Disgust*100)/frame_number (Fear*100)/frame_number
(Happy*100)/frame_number (Sadness*100)/frame_number
(Surprise*100)/frame_number];

                b=bar(app.UIAxes2,X,y);
                xtips1 = b.XEndPoints;
                ytips1 = b.YEndPoints;
                labels1 = string(b.YData);

              text(app.UIAxes2,xtips1,ytips1,labels1,'HorizontalAl
              ignment','center','VerticalAlignment','bottom');
                frame_number = frame_number+1;
                tiempo = toc;
                app.tProcEditField.Value = tiempo;
                if t_framerate < tiempo
                    pause(t_framerate/10);
                else
                    pause(t_framerate - tiempo);
                end
```

La zona resaltada en gris pertenece a la parte del código donde ocurre la clasificación, de cuyo resultado es mostrado en la interfaz en la parte superior derecha actualizándose así la gráfica de porcentaje de aparición.
La parte del código sombreada en azul sin embargo sirve para medir el tiempo de procesamiento de la lógica, y con el tiempo medido tomar decisiones sobre la tasa de reproducción.

Antes de comentar resultados, es necesario mencionar las especificaciones del ordenador donde se ha ejecutado única y exclusivamente esta aplicación. Esta aplicación se ha ejecutado en un ordenador MSI© modelo GL65 9SDK con una CPU Intel© Core i7-9750H @ 2.60GHz de 9ª Generación, de 16 GB de Memoria RAM y Chipset Intel© HM370.

Ahora que ya se conocen las especificaciones, podemos comentar el rango de tiempos de procesamiento que se maneja en esta aplicación. Concretamente, el rango es [0.06s 0.127s], estando la moda en el rango [0.085s 0.105s].

Observando estos resultados, es posible constatar el hecho de que esta aplicación no es capaz de reproducir vídeos en tiempo real que tengan un *framerate* mayor a 9 fotogramas por segundo, por tanto, un vídeo con tasa de fotogramas de 30 fps como ejemplo, tardaría tres veces más de su duración en reproducirse íntegramente.
Esto se debe principalmente a que se realiza un estudio de detección y clasificación de emociones extremadamente exhaustivo, analizando fotograma a fotograma posibles cambios de emoción, operativa que se deberá de estudiar y comprobar que sea viable cambiarla en trabajos futuros como se verá reflejado en el siguiente y último capítulo.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

Durante la realización de este trabajo fin de grado se ha conseguido cumplir el objetivo principal del mismo: desarrollar una aplicación funcional capaz de detectar emociones.

Se ha tratado de un trabajo complejo en cuanto a investigación debido a que se trata de un ámbito de trabajo que ha despertado poco interés hasta el momento, a diferencia de las aplicaciones de reconocimiento facial enfocadas a temas de seguridad, por ejemplo.

El reconocimiento de emociones no se puede decir que sea algo novedoso en el ámbito de investigación puesto que, como se ha visto en el estado del arte, en el año 1978 se formalizó el concepto de *Action Unit* que indirectamente está relacionado con la detección de emociones tal y como previamente se ha comentado. Sin embargo, sí que se trata de un campo poco trabajado por no saber aún una utilidad concreta para ello, mejor dicho, por la no existencia aún de una *killer application*.

Tras hacer la consulta con el Sr. Tadas Baltrušaitis sobre las bases de datos, él mismo intercambió correos electrónicos conmigo para interesarse sobre el tema de la investigación que nos ocupa, llegando a opinar lo siguiente, y cito:

"*I made an explicit choice in OpenFace to recognize facial expressions (Action Units such as smile, brow raise, etc) and behavior descriptors such as head pose and eye gaze instead of emotions (things like happy/sad/etc.). The reason for this is that the former are objective measures what the face is doing, while the latter are much more subjective and open to interpretation and dependent on culture/context/age/gender. I find it helpful to think about facial expressions as the signal, and emotions as the message. While there are commercial tools that predict "emotion" out there, they are often exaggerating their capabilities, as recognizing internal emotion of someone without additional context is almost impossible.*

While there are "rules" for converting facial expressions to a set of "basic emotions" they are just rough guidelines and not very accurate due to the subjectivity and ambiguity of the task. Before you go down that route I would reconsider what exactly you are trying to measure, do you actually need emotions?".

Centrándonos en la parte más importante del mensaje, el Sr. Baltrušaitis constata que su herramienta no está diseñada para detectar ni clasificar emociones debido a que comprende que las emociones son algo no objetivo, basado en la cultura e incluso en edad y género, explicando que existen "reglas" para convertir gestos faciales pero que no son fiables y que son aproximaciones.
A su vez, expresa su opinión sobre las aplicaciones comerciales para detectar emociones, poniendo de manifiesto que pueden llegar a exagerar sus buenos resultados debido a que, según su criterio, una emoción no puede detectarse única y exclusivamente gracias a los gestos faciales, sino que se necesita más contexto.

En cuanto a esto, he de hacer constar que estoy de acuerdo con él en todos los aspectos referidos al reconocimiento de emociones basado en el estudio de características faciales, puesto que el contexto, el lenguaje corporal, la voz e incluso está demostrado que la

temperatura corporal influyen a la hora de detectar emociones, no se puede concebir como completamente fidedigno el reconocimiento de una emoción gracias a ciertos gestos faciales, aunque sí que es un gran indicativo de que puede estar experimentando tal emoción.

5.2 Trabajo futuro

Visto todo lo anterior, no cabe duda alguna de que todavía queda mucho trabajo por delante en temas relacionados directamente con el proyecto realizado. Como ya se ha mencionado, se trata de un campo en el que las posibilidades son muy grandes y gracias a la tecnología en constante progreso se pueden conseguir objetivos prometedores en un futuro no muy lejano.

Referido a este proyecto, y remitiéndome a lo comentado en el apartado anterior, se puede trabajar mucho, hay un horizonte bastante grande de posibilidades debido a que es un tema que nunca antes se había trabajado en un Trabajo de Fin de Grado en el VPULab de la Escuela Politécnica Superior de la UAM.

5.2.1 Mejoras sobre la aplicación desarrollada

En caso de que se pretenda avanzar sobre lo desarrollado, está claro que hay muchas cosas que pueden optimizarse y otras muchas que se pueden implementar para dar con una aplicación mucho más completa.

Como primer avance, y quizás la primera aproximación que se deberá tomar cuando se retome este trabajo, sería sensato repasar el código hecho y optimizarlo con la intención de que las limitaciones que tiene el sistema actualmente se eliminen o en su defecto que se puedan pulir. Esto sobre todo va orientado a que el análisis de los vídeos que se quieran estudiar se pueda realizar a tiempo real, opción que ahora mismo no se ofrece ya que la captura de vídeo y su procesamiento no son inmediatos. Además de que tal y como se ha mencionado anteriormente, la aplicación no es capaz de mostrar los resultados por pantalla en tiempo real debido a que el tiempo de procesamiento no deja reproducir vídeos a velocidad normal a no ser que estos tengan un *framerate* de 9 fps (fotogramas por segundo) o inferior.

Se debería también estudiar la viabilidad de clasificar las emociones no como se ha hecho hasta ahora en este proyecto, que es fotograma a fotograma, sino detectando las emociones con una frecuencia más baja debido a la baja variabilidad de gestos en la cara de una persona, detectando de esta manera la emoción cada 4 o 5 fotogramas, por ejemplo.

Otro tema que podría interesar sería el de añadir una emoción “Neutra”, que en principio es complejo ya que la expresión neutra no es ni mucho menos estándar, gran parte de la población es incapaz de recrear una expresión neutra sin caer en mimetizar la emoción, aunque en intensidad baja, de tristeza o enfado. Se podría buscar la incorporación de esta expresión neutra ya que, obviamente, el ser humano no está constantemente expresando emociones, por tanto, tiene sentido incluirla a pesar de su dificultad.

Se podría intentar juntar los módulos de *OpenFace* y clasificador en un solo conjunto. La ventaja de esto sería el ahorro de pasos a la hora de procesar, no habría que abrir primero *OpenFace* para extraer las características a buscar y luego meterlas en la aplicación por medio de la búsqueda como se hace hasta el momento. Se podría buscar únicamente el

vídeo que se quiere analizar, sin procesarlo previamente como actualmente se realiza, y mediante la implementación de esos dos módulos conjuntos se podría hacer directamente la tarea. Esto sin embargo generaría un cambio de perspectiva del problema, ya que no se podría usar en un principio el espacio de trabajo *OpenFaceOffline* como se ha hecho hasta ahora, se debería de buscar una alternativa.

Por otro lado, sería un gran avance si se pudiera conseguir que el vídeo no fuera, por denominarlo de alguna manera, "pregrabado", sino que se pudiera grabar en el momento en el que se pulse tal opción una vez implementada, ahorrando otro paso más en el proceso actual y haciendo la aplicación más compacta.

Una vez realizadas todas estas tareas, sería interesante que el sistema no funcionara únicamente en ordenador gracias a MatLab, lo ideal sería que funcionara correctamente también fuera de MatLab en sistemas Windows, Linux e incluso Mac como comienzo.

Tras conseguir ese hito, interesaría ir más allá y conseguir su funcionamiento en otros dispositivos como pueden ser tabletas o smartphones, adaptando la aplicación para ser soportada por estos entornos.

En el apartado de conclusiones se constató el hecho de que el análisis de características faciales para detectar emociones no era fiable de forma definitiva, pero sí que gran parte del problema de detectar emociones reside en esa zona del cuerpo.

Entonces, otra vía de estudio que se puede considerar a realizar sería, aparte de optimizar el reconocimiento de emociones mediante gestos faciales como hasta ahora, ahondar más en el problema y tratar de crear una herramienta capaz de realizar la tarea de reconocer emociones gracias al estudio de la cara del sujeto, de la voz y también de la temperatura corporal.

Esta última propuesta no va a modificar la predicción a gran escala, pero teniendo estas dos características en cuenta sí que se podría afirmar que una persona está reflejando o no una emoción con mayor seguridad que estudiando solo la zona facial.

Referencias

[1] D.G. Lowe, “Object recognition from local scale-invariant features”, Proc. of the seventh IEEE international conference on computer vision, vol. 2, pp. 1150-1157, Sept. 1999.

[2] T. Ahonen, A. Hadid, M. Pietikainen, “Face description with local binary patterns: Application to face recognition”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 28(12), 2037-2041, 2006.

[3] J. Sanchez, F. Perronnin, T. Mensink, J. Verbeek. “Image Classification with the Fisher Vector: Theory and Practice”, Research Report, RR-8209, INRIA. 2013..

[4] M.Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 3(1):71-86, March 1991.

[5] M. Chihaoui, A. Elkefi, W. Bellil, C.B. Amar , “A Survey of 2D Face Recognition Techniques”, Computers, 5(4):21, Sept. 2016.

[6] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, A. Ng, “Building high-level features using large scale unsupervised learning”, Proc. of International Conference in Machine Learning 2012

[7] A. Krizhevsky, I. Sutskever, G.E. Hinton, ”Imagenet classification with deep convolutional neural networks”, In *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[8] K. Bahreini, R. Nadolski, W. Westera, “Towards multimodal emotion recognition in E-learning environments “, Interactive Learning Environments, 24(3), 590-605, 2016

[9] B. Allaert, I.M. Bilasco, C. Djeraba. “Micro and macro facial expression recognition using advanced Local Motion Patterns“, IEEE Trans. on Affective Computing (Early Access), 2019 (DOI 10.1109/TAFFC.2019.2949559).

[10] Z. Wu, Y. Wang, G. Pan, “3D Face Recognition Using Local Shape Map”, Proc. of ICIP 2004, pp. 2003-2006.

[11] C. Zhong, Z. Sun, T. Tan, “Robust 3D Face Recognition Using Learned Visual Codebook,” Proc. of CVPR 2007.

[12] A. Bosch, A. Zisserman, X. Munoz, “Representing shape with a spatial pyramid kernel,” Proc. of 6th ACM International Conference on Image and Video Retrieval, 2007, pp. 401–408.

[13] D. Ghimire, J. Lee, “Geometric feature-based facial expression recognition in image sequences using multi-class adaboost and support vector machines,” Sensors, 13(6):7714–7734, 2013.

[14] H. Jung, S. Lee, J. Yim, S. Park, J. Kim, “Joint fine-tuning in deep neural networks for facial expression recognition,” Proc. of ICCV 2015, pp. 2983–2991.

[15] B. Allaert, I. M. Bilasco, C. Djeraba, “Consistent optical flow maps for full and micro facial expression recognition”, Proc. of VISAPP 2017, pp. 235–242.

[16] X. Huang, S. Wang, X. Liu, G. Zhao, X. Feng, M. Pietikainen, “Spontaneous facial micro-expression recognition using discriminative spatiotemporal local binary pattern with an improved integral projection,” Proc. of CVPR 2016.

[17] Y.-J. Liu, J.-K. Zhang, W.-J. Yan, S.-J. Wang, G. Zhao, X. Fu, “A main directional mean optical flow feature for spontaneous microexpression recognition,” IEEE Trans. on Affective Computing, 7(4):299–310, 2016

[18] D. H. Kim, W. Baddar, J. Jang, Y. M. Ro, “Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition”, IEEE Trans. on Affective Computing, 10(23=:223-236, 2017.

[19] Y. Wang, J. Liu, X. Tang, “Robust 3D Face Recognition by Local Shape Difference Boosting”, IEEE Trans. on Pattern Analysis and Machine Intelligence, 32(10):1858-1870, October 2010.
[20] P. Ekman, E. L. Rosenberg, “What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)”. Oxford University Press, USA, 1997.
[21] X. Li, X. Hong, A. Moilanen, X. Huang, T. Pfister, G. Zhao, M. Pietikainen, “Reading hidden emotions: spontaneous micro- ¨ expression spotting and recognition,” Proc. of CVPR 2015, pp. 217–230.
[22] S. Hochreiter, J. Schmidhuber, "Long short-term memory," Neural Comput., 9(8): 1735-1780, 1997.
[23] T. Baltrušaitis, A. Zadeh, Y. Chong Lim, L. P. Morency, “OpenFace 2.0: Facial Behavior Analysis Toolkit”, Proc. of IEEE International Conference on Automatic Face and Gesture Recognition 2018
[24] T. Baltrusaitis, L.-P. Morency, P. Robinson, “Constrained ˇ local neural fields for robust facial landmark detection in the wild”, Proc. of ICCVW 2013.
[25] D. Cristinacce, T. Cootes. “Feature detection and tracking with constrained local models“, Proc. of BMVC 2006.
[26] E. Wood, T. Baltrusaitis, X. Zhang, Y. Sugano, P. Robinson, A. Bulling. “Rendering of eyes for eye-shape registration and gaze estimation”, Proc. of ICCV 2015.
[27] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, “The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression”, Proc. of CVPR 2010, pp. 94-101.
[28] N. Savran, H. Alyüz, O. Dibeklioğlu, B. Çeliktutan, B. Gökberk, B. Sankur, L. Akarun, “Bosphorus Database for 3D Face Analysis”, Proc. of First COST 2101 Workshop on Biometrics and Identity Management (BIOID 2008), 2008.
[29] MathWorks Help Center. “Choose Classifier Options” https://www.mathworks.com/help/stats/choose-a-classifier.html (último acceso Julio 2020)